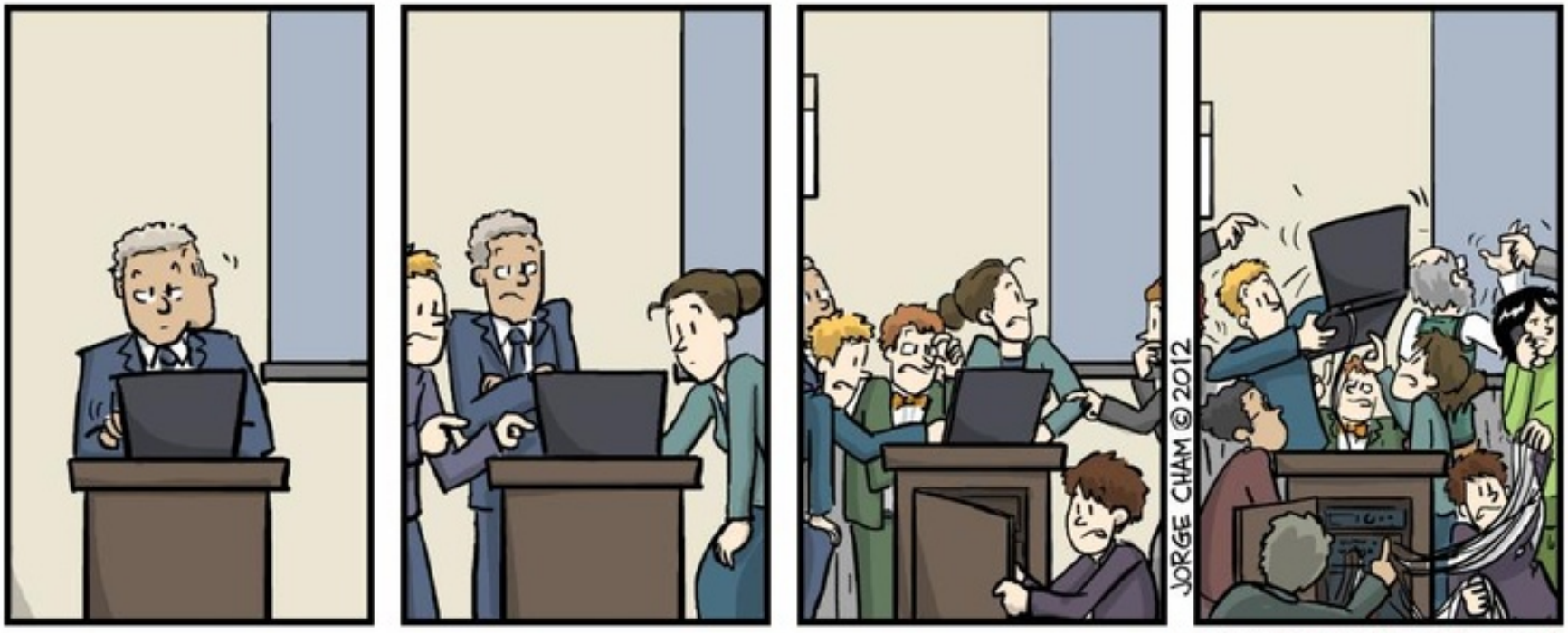
Q: HOW MANY PH.D.'S DOES IT TAKE TO GET A POWERPOINT PRESENTATION TO WORK?



WWW.PHDCOMICS.COM

ANSWER: (n+1)

WHERE n = THE NUMBER OF ACADEMICS IN THE ROOM WHO THINK THEY KNOW HOW TO FIX IT, AND 1 = THE PERSON WHO FINALLY CALLS THE A/V TECHNICIAN.

Uncertainty reasoning and machine learning

A Few Applications

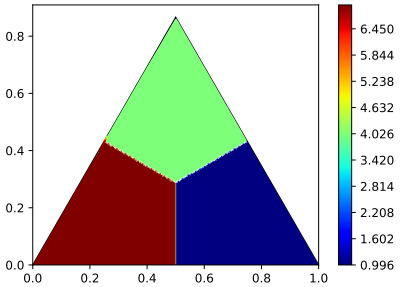
Vu-Linh Nguyen

Chaire de Professeur Junior, Laboratoire Heudiasyc
Université de technologie de Compiègne

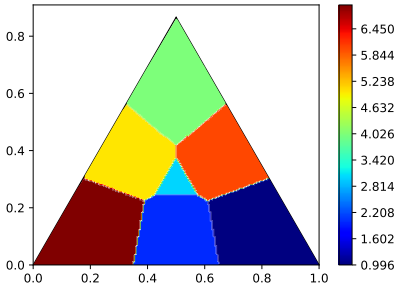
AOS4 master courses

Optimal Decision Rules

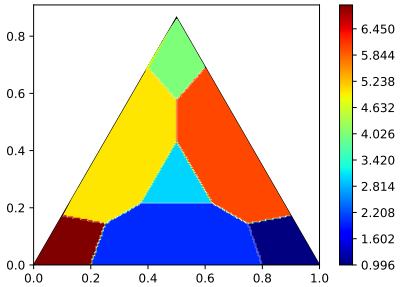
Probabilistic approach



0/1 loss



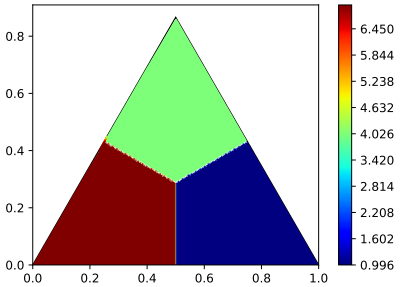
$u_{1.6}$



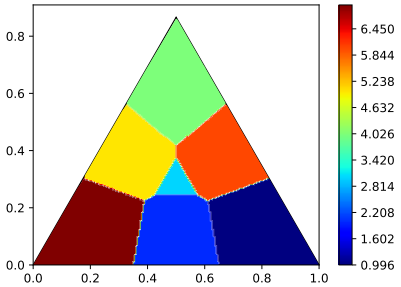
$u_{2.2}$

Optimal Decision Rules

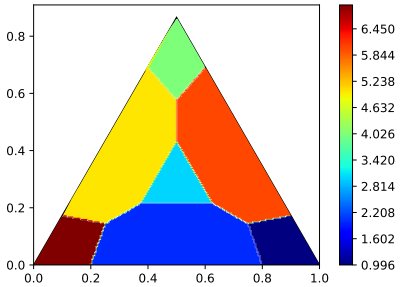
Probabilistic approach



0/1 loss

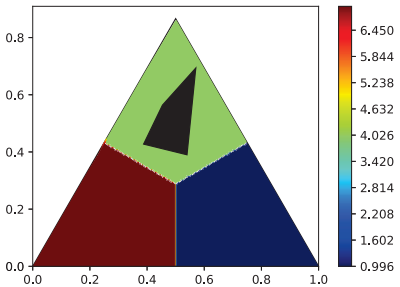


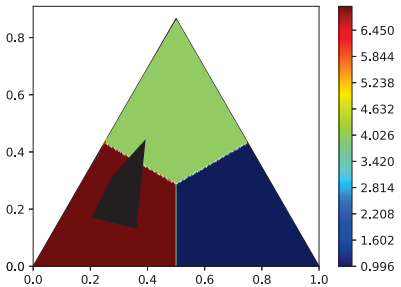
$u_{1.6}$

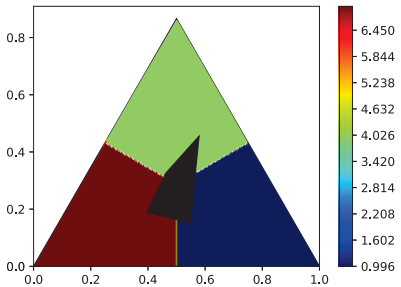


$u_{2.2}$

Credal approach







Objectives

After this lecture, students should be able to describe (a few)

- probabilistic and credal classifiers
- and how to use them to make singleton and set-valued predictions,
- and their (potential) applications.

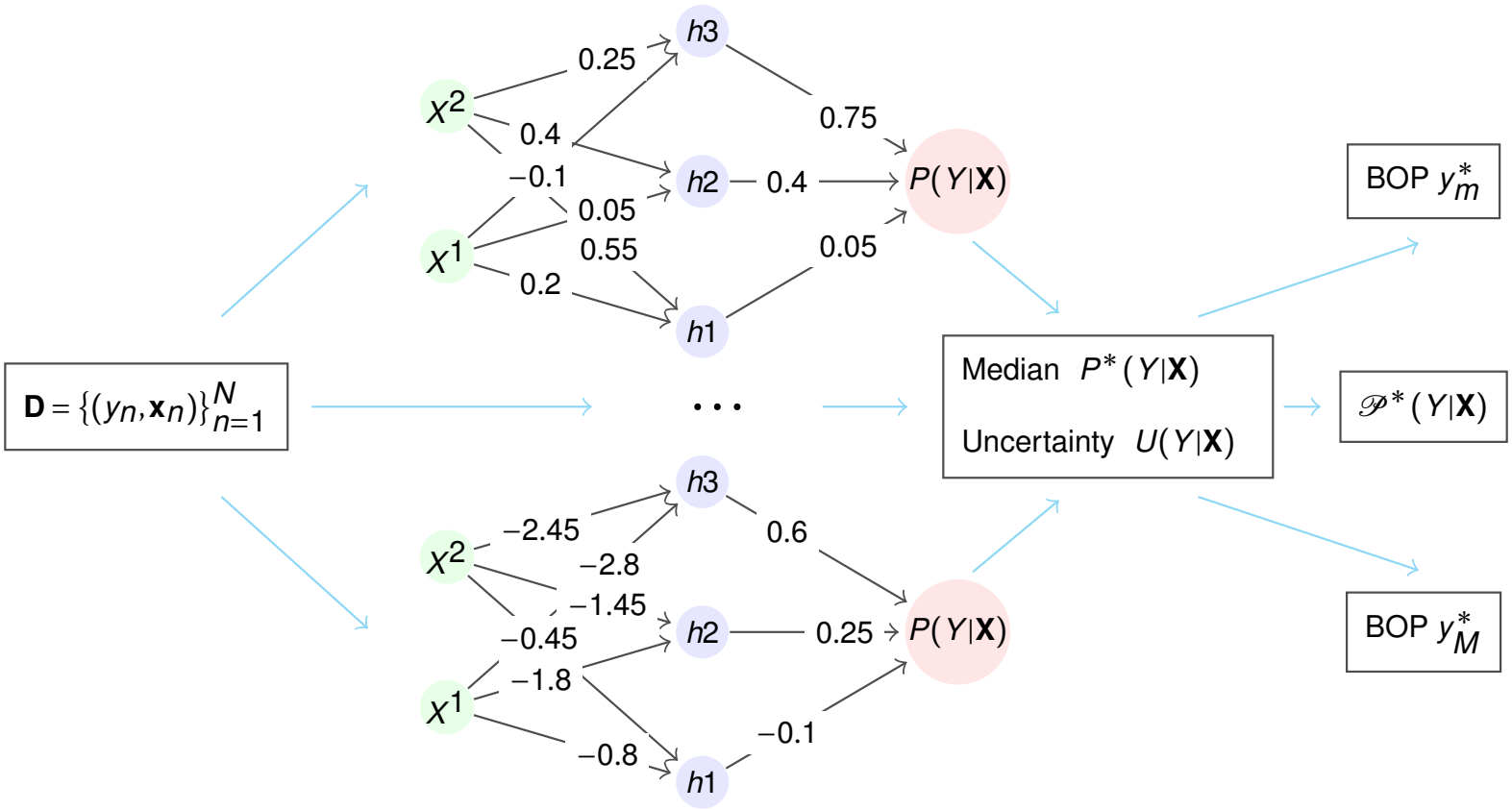
Outline

- Credal ensembling in multi-class classification
  - A median classifier: Learning and inference
  - A credal classifier: Learning and inference
  - Applications in machine learning
- Compact deep ensembles
- Other applications

A formal framework [4]

Basic setup:

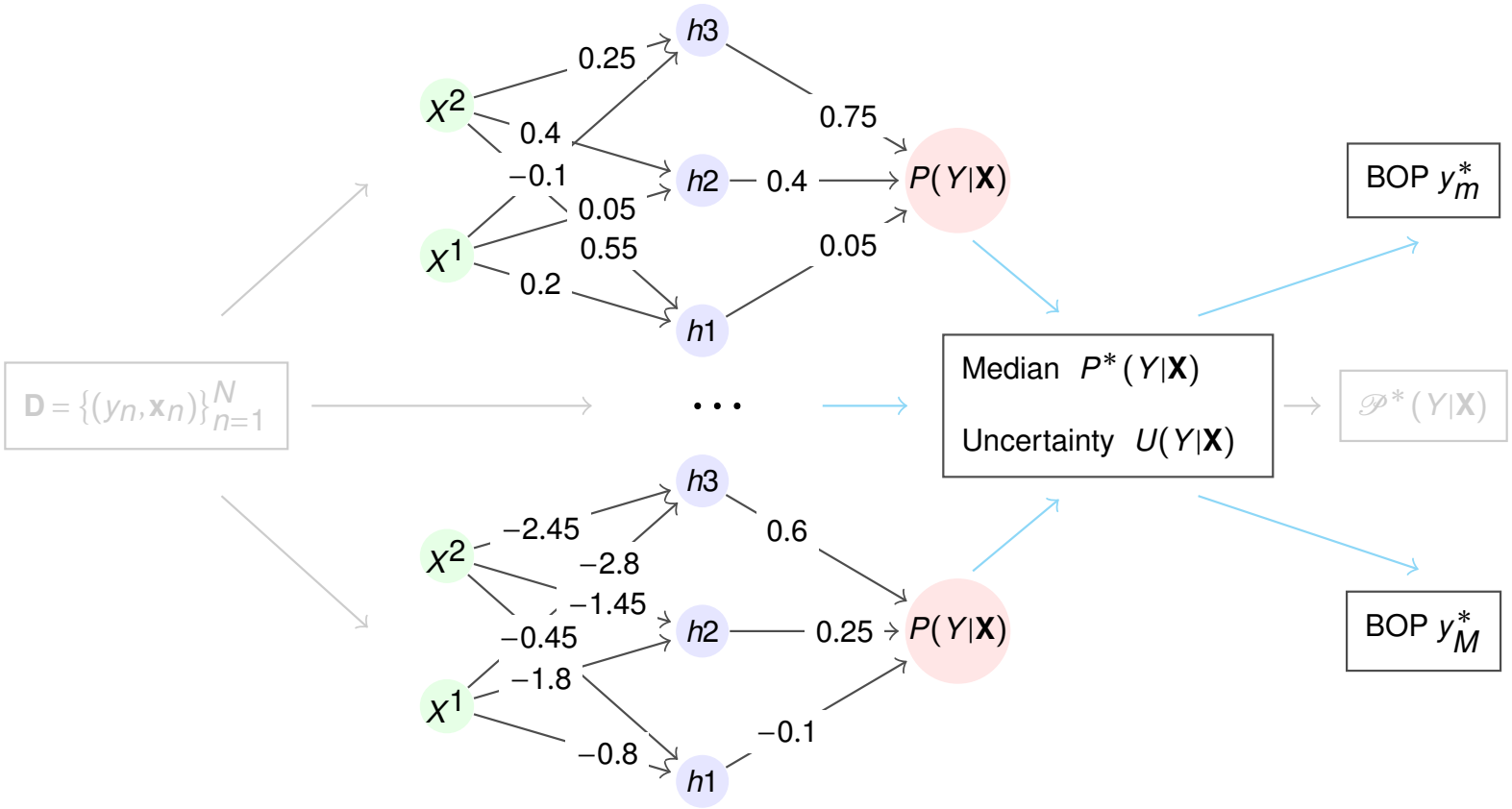
- Features $(X^1,\ldots,X^P)$ and a class variable Y
- An finite output space $\mathcal{Y} = \{y^1,\ldots,y^C\}$



Outline

- Credal ensembling in multi-class classification
  - A median classifier: Learning and inference
  - A credal classifier: Learning and inference
  - Applications in machine learning
- Compact deep ensembles
- Other applications

A median classifier and its predictions [4]



Compute a median classifier

Basic setting:

- An ensemble $\mathbf{H} := \{\mathbf{h}^m | m \in [M] := \{1, \ldots, M\}\}$ is made available
- A specified statistical distance d between distributions

A median classifier minimizes the average expected distance:

$$\mathbf{h}_d \in \underset{\mathbf{h} \in \mathscr{H}}{\operatorname{argmin}} \mathbf{E}\left[\sum_{m=1}^{M} d(\mathbf{h}, \mathbf{h}^m)\right] = \underset{\mathbf{h} \in \mathscr{H}}{\operatorname{argmin}} \int_{\mathbf{x} \in \mathscr{X}} \left[\sum_{m=1}^{M} d(\mathbf{h}(\mathbf{x}), \mathbf{h}^m(\mathbf{x}))\right] d\mathbf{x}.$$

Compute a median classifier

Basic setting:

- An ensemble $\mathbf{H} := \{\mathbf{h}^m | m \in [M] := \{1, \ldots, M\}\}$ is made available
- A specified statistical distance d between distributions

A median classifier minimizes the average expected distance:

$$\mathbf{h}_d \in \operatorname*{argmin}_{\mathbf{h} \in \mathscr{H}} \mathbf{E}\left[\sum_{m=1}^{M} d(\mathbf{h}, \mathbf{h}^m)\right] = \operatorname*{argmin}_{\mathbf{h} \in \mathscr{H}} \int_{\mathbf{x} \in \mathscr{X}} \left[\sum_{m=1}^{M} d(\mathbf{h}(\mathbf{x}), \mathbf{h}^m(\mathbf{x}))\right] d\mathbf{x}.$$

If no constraint on $\mathscr{H}$, $\mathbf{h}_d$ can be defined in an instance-wise manner:

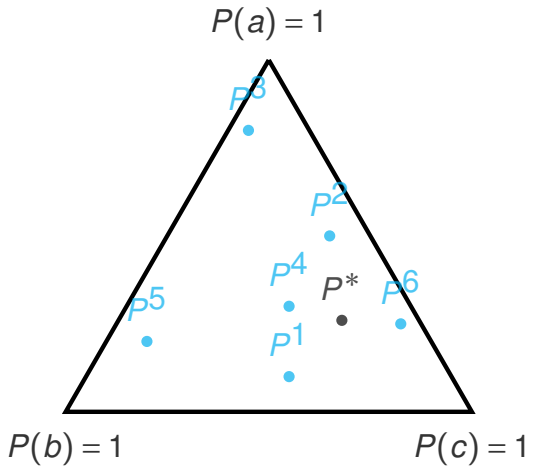
$$\mathbf{h}_d(\mathbf{x}) \in \operatorname*{argmin}_{\mathbf{h}(\mathbf{x}) \in \Delta^K} \sum_{m=1}^{M} d(\mathbf{h}(\mathbf{x}), \mathbf{h}^m(\mathbf{x})). \tag{1}$$

Compute a median classifier (cont.)

For each **x**, dropping **x** and denoting $P = \mathbf{h}$ give

$$P_d \in \operatorname*{argmin}_{P \in \Delta^K} \sum_{m=1}^{M} d(P, P^m). \qquad (2)$$

Examples of d are squared Euclidean distance (sE), L_1 distance, and KL divergence.



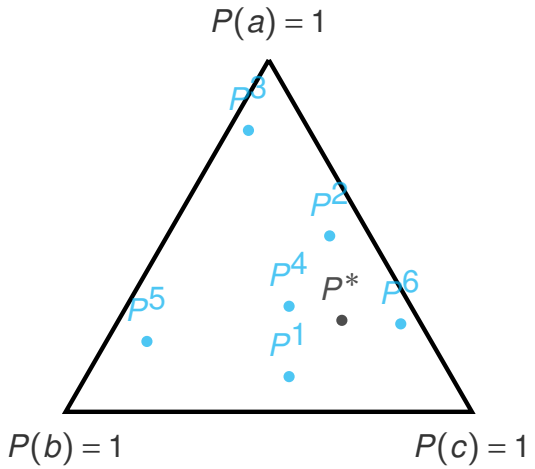
Ensemble **H** and P_{sE}

Compute a median classifier (cont.)

For each **x**, dropping **x** and denoting $P = \mathbf{h}$ give

$$P_d \in \underset{P \in \Delta^K}{\operatorname{argmin}} \sum_{m=1}^{M} d(P, P^m). \qquad (2)$$

Examples of d are squared Euclidean distance (sE), L_1 distance, and KL divergence.



Ensemble **H** and P_{sE}

For any convex distance d:

- The convex optimization problem (2) can be solved using any solver.
- Close-form solution P_{sE} = averaging the distributions class-wise.

Bayes-optimal predictions

Basic set (instance-wise manner):

- The median distribution P_d is given.
- A the higher the better utility $u : \mathscr{Y} \times \mathscr{Y} \longmapsto \mathbb{R}_+$

A **Bayes-optimal prediction (BOP) of** u is

$$y_u^{P_d} \in \operatorname*{argmax}_{y' \in \mathscr{Y}} \mathbf{E}\left[u(y', y)\right] = \operatorname*{argmax}_{y' \in \mathscr{Y}} \sum_{y \in \mathscr{Y}} u(y', y) P_d(y)\,. \tag{3}$$

Bayes-optimal predictions

Basic set (instance-wise manner):

- The median distribution P_d is given.
- A the higher the better utility $u : \mathscr{Y} \times \mathscr{Y} \longmapsto \mathbb{R}_+$

A **Bayes-optimal prediction (BOP) of** u is

$$y_u^{P_d} \in \operatorname*{argmax}_{y' \in \mathscr{Y}} \mathbf{E}\left[u(y', y)\right] = \operatorname*{argmax}_{y' \in \mathscr{Y}} \sum_{y \in \mathscr{Y}} u(y', y) P_d(y). \tag{3}$$

Commonly used utilities, such as $0/1$ and cost-sensitive accuracies:

- Find a BOP (10) takes from $O(K)$ to $O(K^2)$
- A BOP $y_u^{P_d}$ (10) of $0/1$ accuracy = a most probable class

Bayes-optimal set-valued predictions

Basic set (instance-wise manner):

- The median distribution P_d is given.
- A the higher the better utility $U : \mathscr{Y} \times 2^{\mathscr{Y}} \longmapsto \mathbb{R}_+$

A **Bayes-optimal prediction (BOP) of** U is

$$S_U^{P_d} \in \underset{S' \subset \mathscr{Y}}{\operatorname{argmax}} \mathbf{E}\left[U(S', y)\right] = \underset{S' \subset \mathscr{Y}}{\operatorname{argmax}} \sum_{y \in \mathscr{Y}} U(S', y) P_d(y). \tag{4}$$

Bayes-optimal set-valued predictions

Basic set (instance-wise manner):

- The median distribution P_d is given.
- A the higher the better utility $U : \mathcal{Y} \times 2^{\mathcal{Y}} \longmapsto \mathbb{R}_+$

A **Bayes-optimal prediction (BOP) of** U is

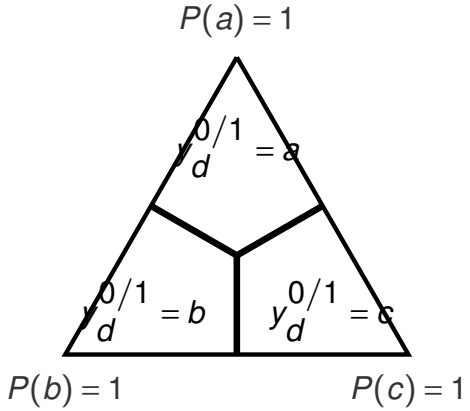
$$S_U^{P_d} \in \operatorname*{argmax}_{S' \subset \mathcal{Y}} \mathbf{E}\left[U(S', y)\right] = \operatorname*{argmax}_{S' \subset \mathcal{Y}} \sum_{y \in \mathcal{Y}} U(S', y) P_d(y). \tag{4}$$

Commonly used utilities, such as utility-discounted accuracies:

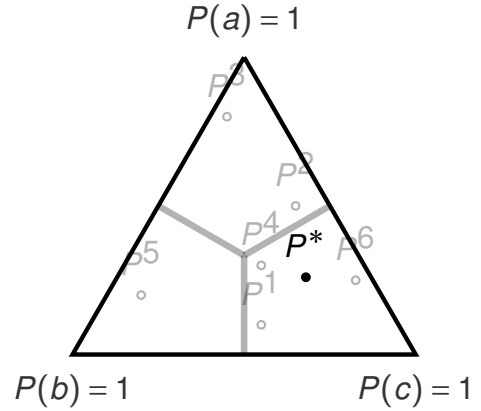
$$U(S', y) = \frac{1}{g(|S'|)} [\![y \in S']\!], \tag{5}$$

- Find a BOP S_d^U (15) takes $O(K \log(K))$.
- A BOP $S_U^{P_d}$ (15) consists of the most probable classes on P_d.

Probabilistic uncertainty scores

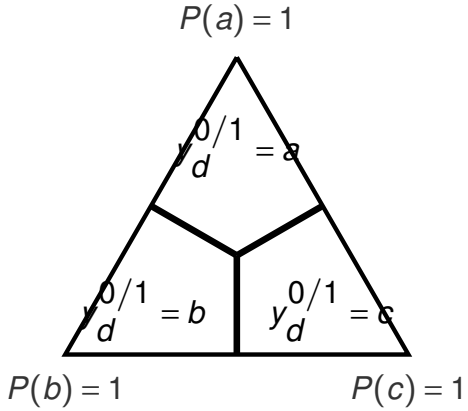


Regions $\Delta_{\mathbf{x}}^{\text{st}}$ for different $y_d^{0/1}$

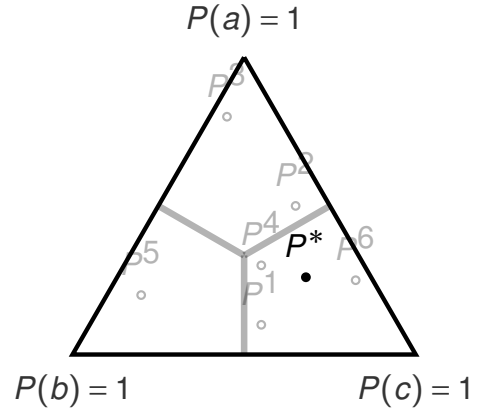


Ensemble $\mathbf{H}$ and P_{sE}

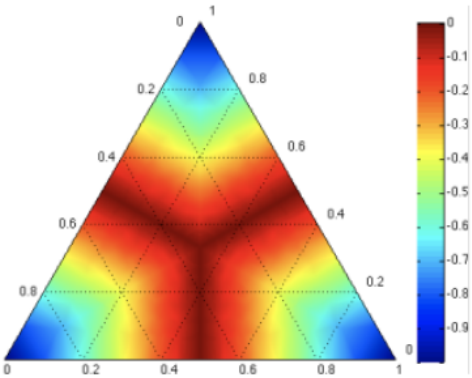
Probabilistic uncertainty scores



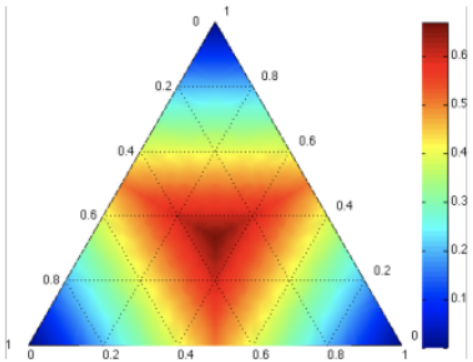
Regions $\Delta_{\mathbf{x}}^{\text{st}}$ for different $y_d^{0/1}$



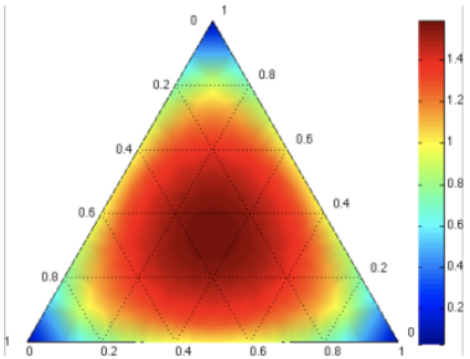
Ensemble **H** and P_{sE}



(a) Smallest margin (↑)



(b) Least confidence (↓)



(c) Entropy (↓)

Heatmaps illustrating the **behavior of probabilistic uncertainty scores**

Probabilistic uncertainty scores (Cont.)

Smallest margin (↑) is defined as

$$S_{\mathrm{SM}}(P_d) = P_d\left(y^{\mathrm{st}}\right) - P_d\left(y^{\mathrm{nd}}\right). \tag{6}$$

Example: A classification problem with $\mathscr{Y} = \{a, b, c\}$:

	$\mathbf{x}_1$	$\mathbf{x}_2$
	50→(0.6, 0.4, 0.0) 50→(0.0, 0.4, 0.6)	100→(0.3, 0.4, 0.3)
P_{sE}	(0.3, 0.4, 0.3)	
S_{SM} (↑)	0.1	
	Should we consider $\mathbf{x}_1$ and $\mathbf{x}_2$ the same?	

Probabilistic uncertainty scores (Cont.)

Smallest margin (↑) is defined as

$$S_{\mathrm{SM}}(P_d) = P_d\left(y^{\mathrm{st}}\right) - P_d\left(y^{\mathrm{nd}}\right). \tag{6}$$

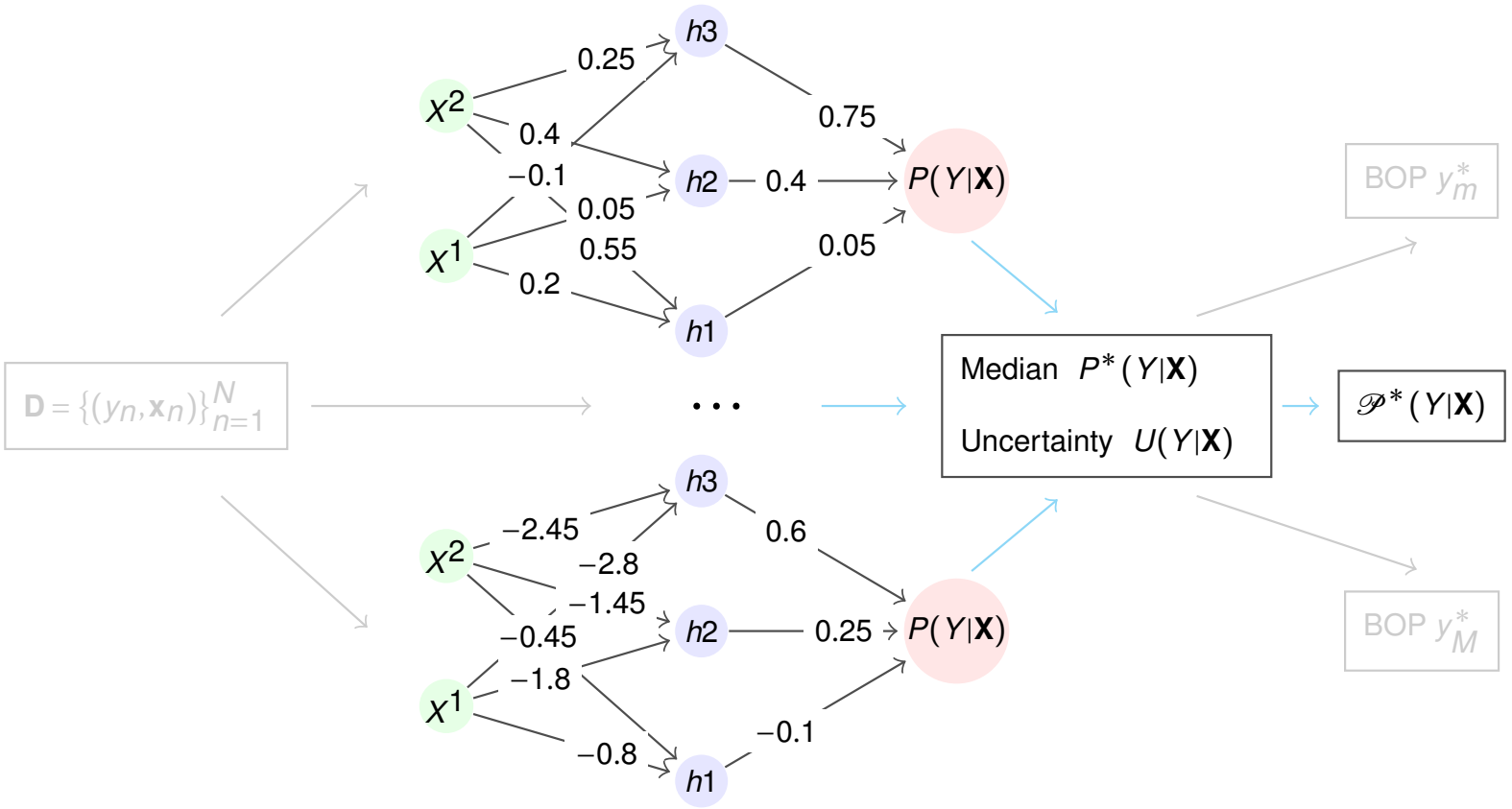
Example: A classification problem with $\mathscr{Y} = \{a, b, c\}$:

	$\mathbf{x}_3$	$\mathbf{x}_4$
	80→(1.0, 0.0, 0.0) 20→ (0.0, 1.0, 0.0)	100→(0.8, 0.2, 0.0)
P_{sE}	(0.8, 0.2, 0.0)	
S_{SM} (↑)	0.6	
	Should we consider $\mathbf{x}_3$ and $\mathbf{x}_4$ the same?	

Outline

- Credal ensembling in multi-class classification
  - A median classifier: Learning and inference
  - A credal classifier: Learning and inference
  - Applications in machine learning
- Compact deep ensembles
- Other applications

A credal classifier and its predictions [4]



For any query instance, once $\mathscr{P}^*(\mathscr{Y}|\mathbf{x})$ is estimated:

- IP decision rules can be called to make set-valued predictions
- uncertainty scores defined for credal sets can be computed.

Estimate a credal classifier

Each credal classifier $\mathbf{CH}_\alpha^d$ is defined in a point-wise manner:

$$\mathbf{CH}_\alpha^d := \left\{ P := \sum_{m=1}^{M_\alpha} \gamma_m P^{(m)} | \gamma_m \geq 0, m \in [M_\alpha], \sum_{m=1}^{M_\alpha} \gamma_m = 1 \right\}, \tag{7}$$

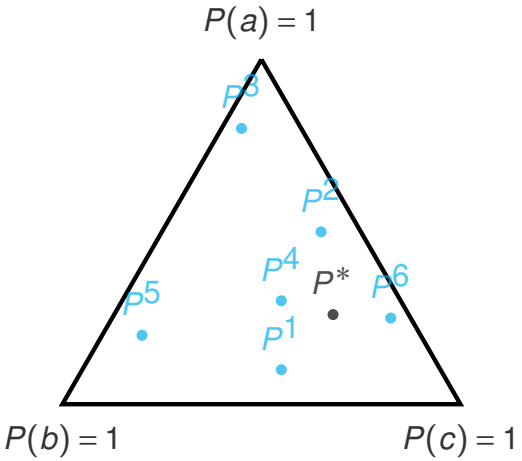
where $P^{(m)}$ is the m-th closet point to P_d **according to the distance** d.

Estimate a credal classifier

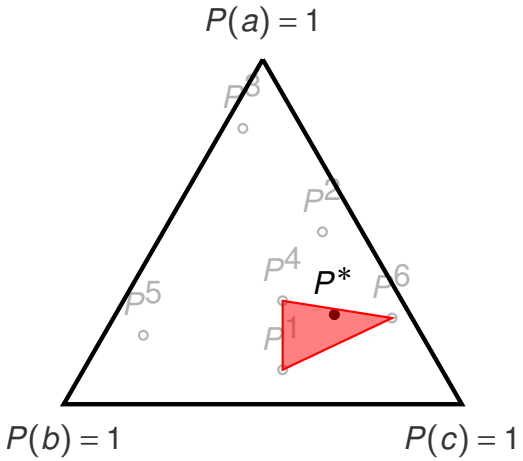
Each credal classifier $\mathbf{CH}_\alpha^d$ is defined in a point-wise manner:

$$\mathbf{CH}_\alpha^d := \left\{ P := \sum_{m=1}^{M_\alpha} \gamma_m P^{(m)} | \gamma_m \geq 0, m \in [M_\alpha], \sum_{m=1}^{M_\alpha} \gamma_m = 1 \right\}, \qquad (7)$$

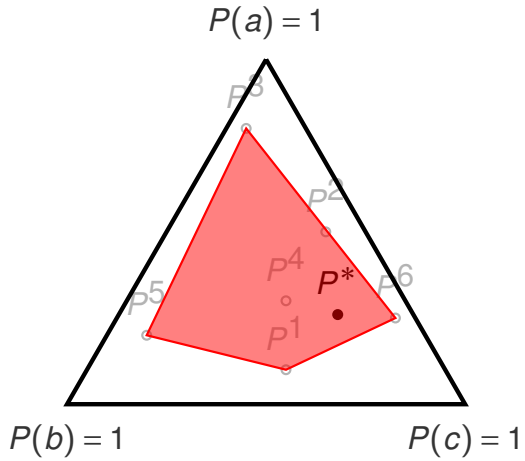
where $P^{(m)}$ is the m-th closet point to P_d **according to the distance** d.



Ensemble **H** and P^*_{sE}



Credal set $\mathbf{CH}^{\text{sE}}_{0.5}$



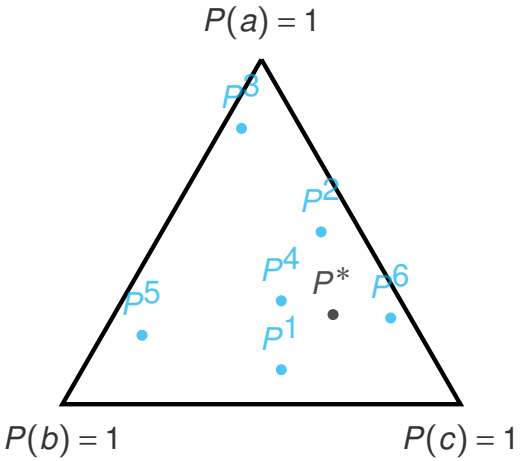
Credal set $\mathbf{CH}^{\text{sE}}_{0}$

Estimate a credal classifier

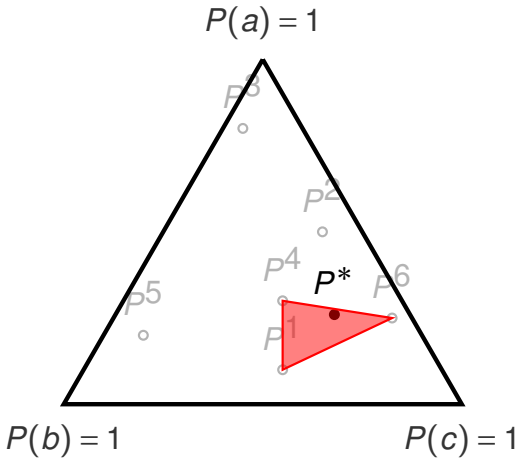
Each credal classifier $\mathbf{CH}_\alpha^d$ is defined in a point-wise manner:

$$\mathbf{CH}_\alpha^d := \left\{ P := \sum_{m=1}^{M_\alpha} \gamma_m P^{(m)} | \gamma_m \geq 0, m \in [M_\alpha], \sum_{m=1}^{M_\alpha} \gamma_m = 1 \right\}, \qquad (7)$$

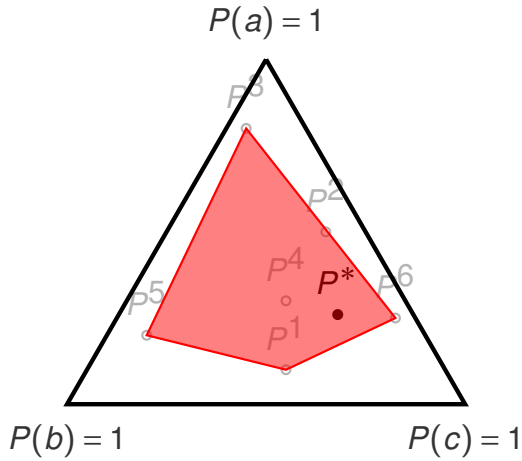
where $P^{(m)}$ is the m-th closet point to P_d **according to the distance** d.



Ensemble **H** and P^*_{sE}



Credal set $\mathbf{CH}^{\text{sE}}_{0.5}$

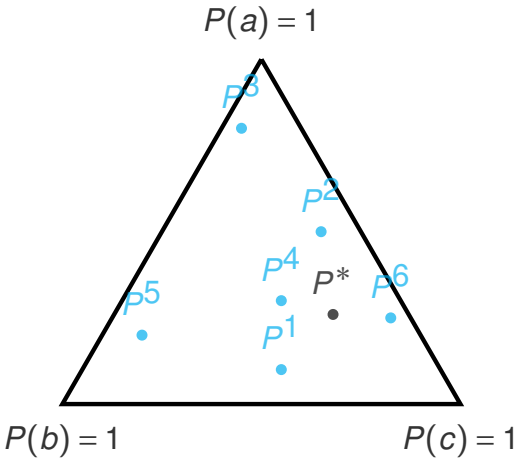


Credal set $\mathbf{CH}^{\text{sE}}_{0}$

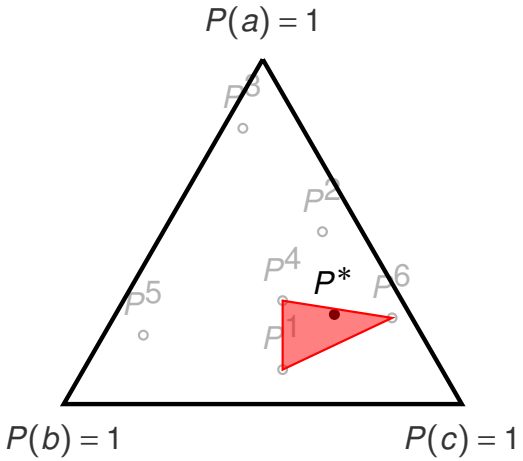
The hyperparameter α^* ← nested cross validation or a validation set.

Optimal set-valued predictions under IP decision rules

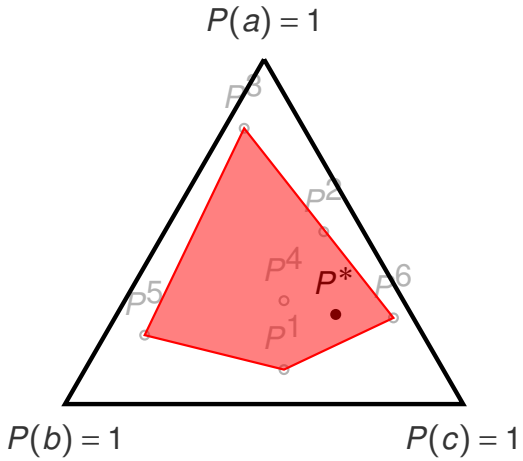
Basic set (instance-wise manner): The credal set $\mathbf{CH}^{d}_{\alpha^*}$ is given.



Ensemble **H** and P_{sE}



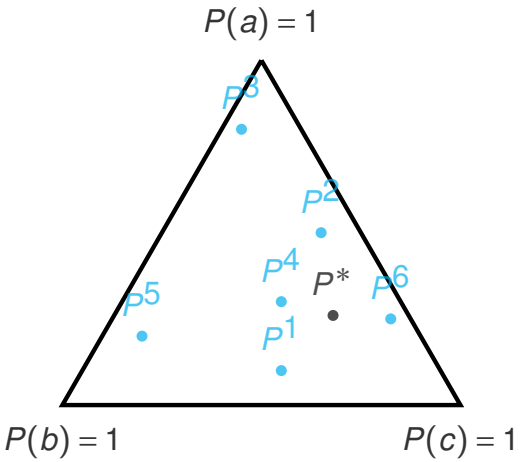
Credal set $\mathbf{CH}^{\text{sE}}_{0.5}$



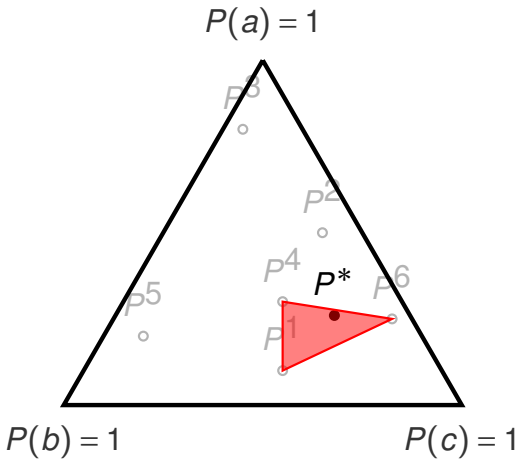
Credal set $\mathbf{CH}^{\text{sE}}_{0}$

Optimal set-valued predictions under IP decision rules

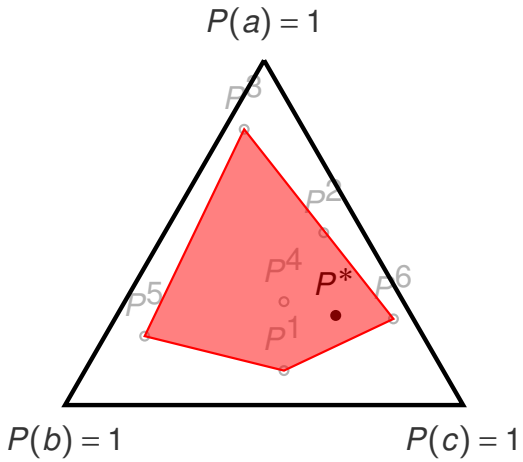
Basic set (instance-wise manner): The credal set $\mathbf{CH}^{d}_{\alpha^*}$ is given.



Ensemble **H** and P_{sE}



Credal set $\mathbf{CH}^{\text{sE}}_{0.5}$



Credal set $\mathbf{CH}^{\text{sE}}_{0}$

- Any IP decision rule $R_{\text{IP}} : 2^{\Delta^K} \longmapsto 2^{\mathcal{Y}}$ can be applied.
- Any related algorithmic solutions can be leveraged.

Optimal set-valued predictions under IP decision rules (Cont.)

Basic set (instance-wise manner):

- The credal set $\mathbf{CH}^d_{\alpha^*}$ is given.
- A the higher the better utility $u : \mathscr{Y} \times \mathscr{Y} \longmapsto \mathbb{R}_+$

E-admissibility under u:

- A class y is E-admissible if there exist $P \in \mathbf{CH}^d_{\alpha^*}$ so that $y = y^u$.
- This can be checked by solving a linear program.

Optimal set-valued predictions under IP decision rules (Cont.)

Basic set (instance-wise manner):

- The credal set $\mathbf{CH}^d_{\alpha^*}$ is given.
- A the higher the better utility $u : \mathscr{Y} \times \mathscr{Y} \longmapsto \mathbb{R}_+$

E-admissibility under u:

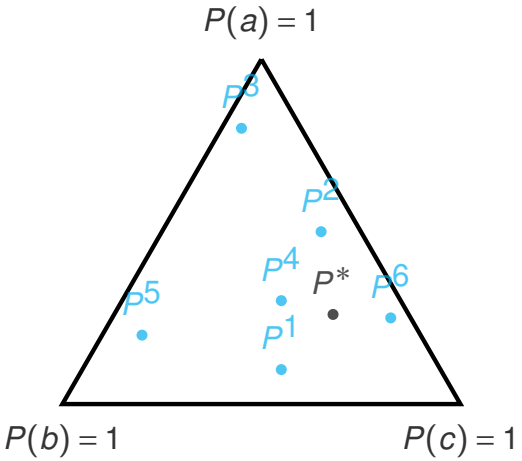
- A class y is E-admissible if there exist $P \in \mathbf{CH}^d_{\alpha^*}$ so that $y = y^u$.
- This can be checked by solving a linear program.

Maximality under u:

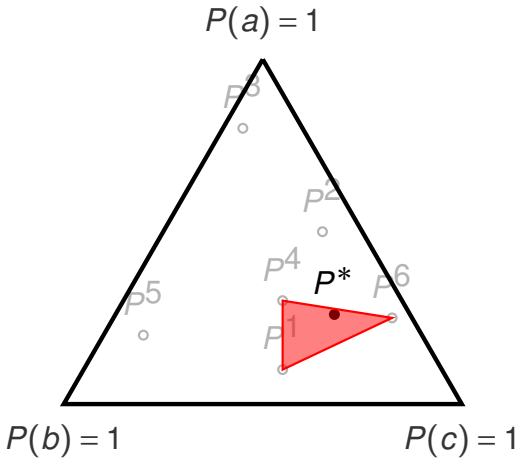
- A class y is maximal if there doesn't exist $y' \neq y$ such that y' dominates y on all $P \in \mathbf{CH}^d_{\alpha^*}$ (w.r.t. u).
- This can be checked by solving $K - 1$ linear programs.
- We can also enumerate all the distributions P^m, $m \in [M]$.

Credal set-based uncertainty scores

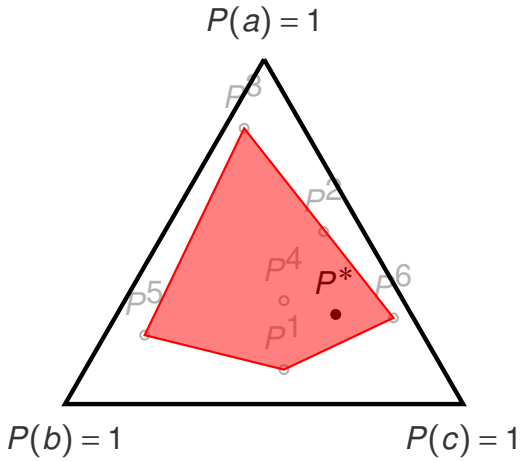
Basic set (instance-wise manner): The credal set $\mathbf{CH}^{d}_{\alpha^*}$ is given.



Ensemble **H** and P_{sE}



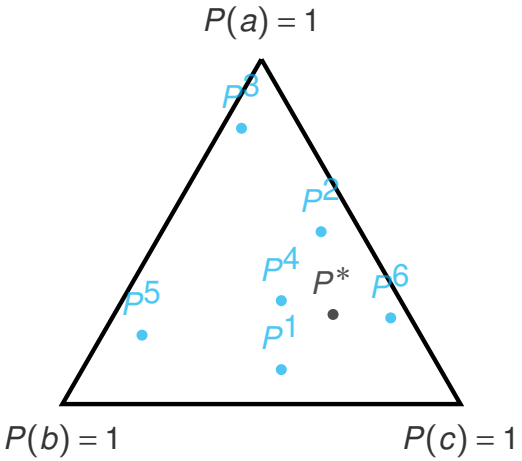
Credal set $\mathbf{CH}^{\text{sE}}_{0.5}$



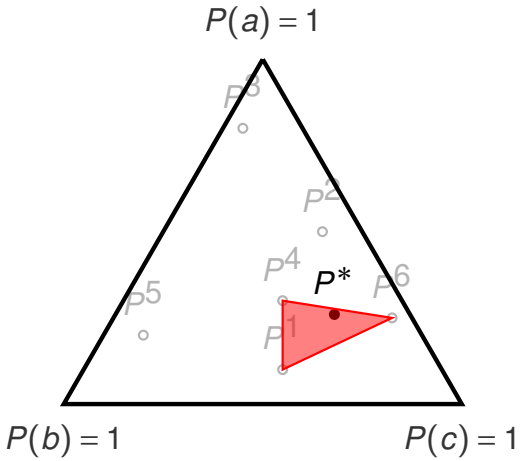
Credal set $\mathbf{CH}^{\text{sE}}_{0}$

Credal set-based uncertainty scores

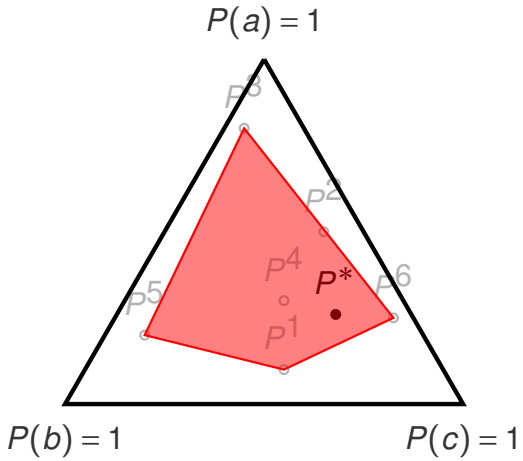
Basic set (instance-wise manner): The credal set $\mathbf{CH}^{d}_{\alpha^*}$ is given.



Ensemble **H** and P_{sE}



Credal set $\mathbf{CH}^{\text{sE}}_{0.5}$



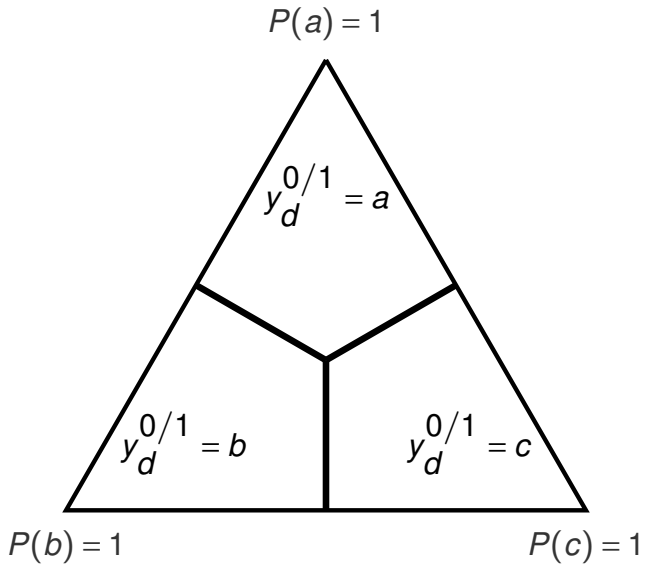
Credal set $\mathbf{CH}^{\text{sE}}_{0}$

- Any credal set-based uncertainty score can be used.
- Any related algorithmic solutions can be leveraged.

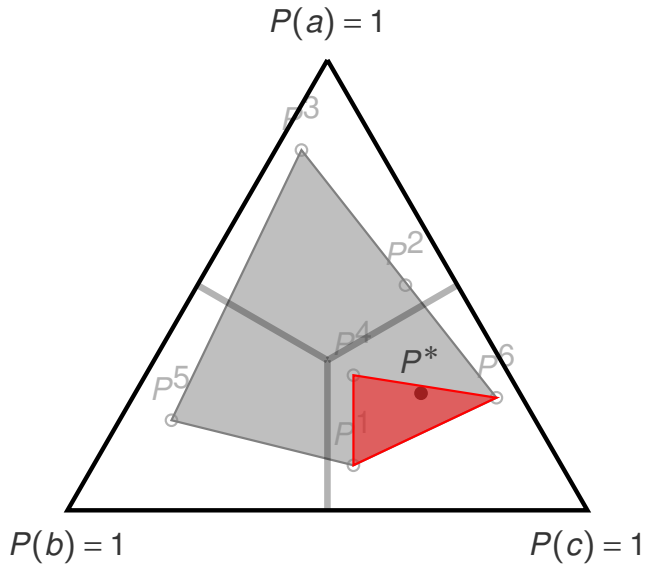
Credal set-based uncertainty scores (Cont.)

Decision-related uncertainty scores:

- How certain the ensemble **H** is about y_d^u?
- How consensus of the ensemble members is about y_d^u?



Regions $\Delta_{\mathbf{x}}^{\text{st}}$ for different $y_d^{0/1}$



Credal set $\mathbf{CH}_0^{\text{sE}}(\mathbf{x})$ with P_{sE}

Decision-related uncertainty scores

Basic setting (instance-wise manner):

- The median distribution P_d is given.
- The predictions $\{P^m | m \in [M]\}$ are given.
- A probabilistic uncertainty score $S : \Delta^K \longmapsto \mathbb{R}$ is given.

Decision-related uncertainty scores

Basic setting (instance-wise manner):

- The median distribution P_d is given.
- The predictions $\{P^m | m \in [M]\}$ are given.
- A probabilistic uncertainty score $S : \Delta^K \longmapsto \mathbb{R}$ is given.

A **decision-related uncertainty** version of S is (defined as its empirical expectation)

$$\mathrm{RS}(P_d^u) := \frac{1}{M+1} \left(\sum_{m=1}^{M} \left[\!\left[P^m \in \mathbf{CH}_{y_d^u}^d \right]\!\right] S(P^m) + S(P_d) \right), \tag{8}$$

where $\left[\!\left[P^m \in \mathbf{CH}_{y_d^u}^d \right]\!\right] = 1$ implies y_d^u is a best solution on P^m under u.

Decision-related uncertainty scores (Cont.)

Smallest margin (↑) is defined as

$$S_{\mathrm{SM}}(P_d) = P_d\left(y^{\mathrm{st}}\right) - P_d\left(y^{\mathrm{nd}}\right). \tag{9}$$

Example: A classification problem with $\mathscr{Y} = \{a, b, c\}$:

	$\mathbf{x}_1$	$\mathbf{x}_2$
	50→(0.6, 0.4, 0.0) 50→(0.0, 0.4, 0.6)	100→(0.3, 0.4, 0.3)
P_{sE}	(0.3, 0.4, 0.3)	
S_{SM} (↑)	0.1	
$\mathrm{RS}_{\mathrm{SM}}$ (↑)	0.0	0.1

Decision-related uncertainty scores (Cont.)

Smallest margin (↑) is defined as

$$S_{\mathrm{SM}}(P_d) = P_d\left(y^{\mathrm{st}}\right) - P_d\left(y^{\mathrm{nd}}\right). \tag{9}$$

Example: A classification problem with $\mathscr{Y} = \{a, b, c\}$:

	$\mathbf{x}_3$	$\mathbf{x}_4$
	80→(1.0, 0.0, 0.0) 20→(0.0, 1.0, 0.0)	100→(0.8, 0.2, 0.0)
P_{sE}	(0.8, 0.2, 0.0)	
S_{SM} (↑)	0.6	
$\mathrm{RS}_{\mathrm{SM}}$ (↑)	0.798	0.6

Should we put weights on the impact of ensemble members?

Outline

- Credal ensembling in multi-class classification
  - A median classifier: Learning and inference
  - A credal classifier: Learning and inference
  - Applications in machine learning
- Compact deep ensembles
- Other applications

Experimental setting

Basic setting:

- Use random forests of cardinality 100 as the ensembles
- Follow a 10-cross validation protocol.
- Use hyperparameter $\alpha^* \leftarrow$ nested 10 fold cross validation

Assess the impact of P_{sE}, P_{L1} and P_{KL} on

- the clean version of the data sets
- noisy version (randomly flip the class of 25% of training instances)

Experimental setting

Basic setting:

- Use random forests of cardinality 100 as the ensembles
- Follow a 10-cross validation protocol.
- Use hyperparameter $\alpha^* \leftarrow$ nested 10 fold cross validation

Assess the impact of P_{sE}, P_{L1} and P_{KL} on

- the clean version of the data sets
- noisy version (randomly flip the class of 25% of training instances)

Once credal set $\text{CH}^d_{\alpha^*}$ is computed, it is used to

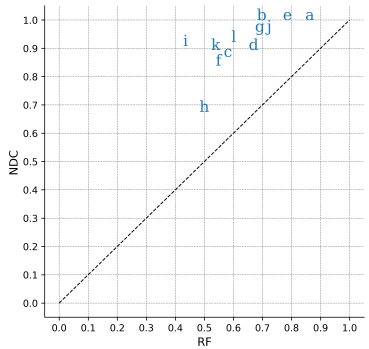
- find the set-valued prediction under the E-admissibility rule.

Results on clean data sets: U_{65} scores (in %) [4]

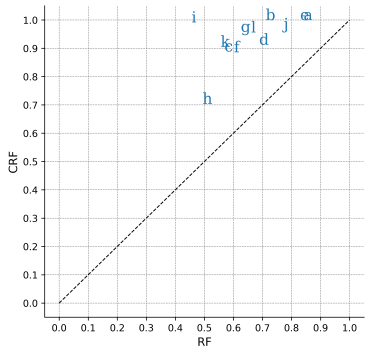
Data set: (N,P,K)	NDC	SQE-E	L1-E	KL-E	CRF	$\mathbf{CH}_0$
eco.: (336,7,8)	85.51	86.07	85.81	**87.07**	84.46	43.60
der.: (358,34,6)	97.18	97.05	97.22	**98.59**	96.19	51.74
lib.: (360, 90, 15)	76.58	73.35	75.24	**79.41**	73.45	14.60
vow.: (990, 10, 11)	86.63	86.35	87.65	**92.35**	82.68	17.75
win.: (1599, 11, 6)	**68.66**	68.32	68.39	68.63	67.35	36.53
seg.: (2300, 19, 7)	97.17	97.12	96.99	**97.64**	96.73	71.00

Results on clean data sets: U_{65} scores (in %) [4]

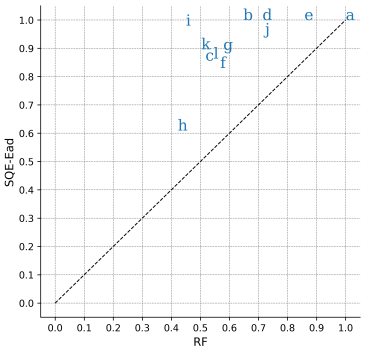
Data set: (N,P,K)	NDC	SQE-E	L1-E	KL-E	CRF	$\mathbf{CH}_0$
eco.: (336,7,8)	85.51	86.07	85.81	**87.07**	84.46	43.60
der.: (358,34,6)	97.18	97.05	97.22	**98.59**	96.19	51.74
lib.: (360, 90, 15)	76.58	73.35	75.24	**79.41**	73.45	14.60
vow.: (990, 10, 11)	86.63	86.35	87.65	**92.35**	82.68	17.75
win.: (1599, 11, 6)	**68.66**	68.32	68.39	68.63	67.35	36.53
seg.: (2300, 19, 7)	97.17	97.12	96.99	**97.64**	96.73	71.00



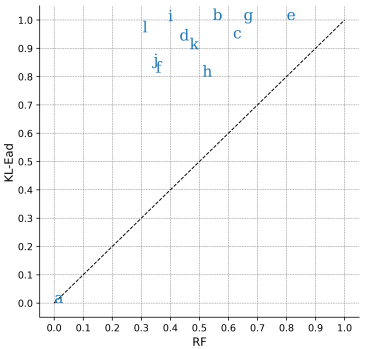
(a) NDC vs RF



(b) CRF vs RF



(e) SQE-Ead vs RF



(f) KL-Ead vs RF

Correctness of cautious predictors (vertical) vs accuracy of RF (horizontal)

Experimental setting [4]

Basic setting:

- Randomly flip the class of 25% of training instances
- Using random forests of cardinality 100 as the ensembles
- The median P_{sE} is employed
- Assess smallest margin S_{SM} (↑) and $\mathbf{RS}_{SM}$ (↑)

Experimental setting [4]

Basic setting:

- Randomly flip the class of 25% of training instances
- Using random forests of cardinality 100 as the ensembles
- The median P_{sE} is employed
- Assess smallest margin S_{SM} (↑) and $\mathbf{RS}_{SM}$ (↑)

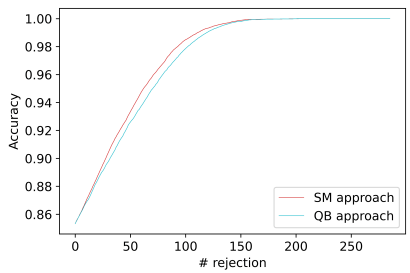
Budget based rejection protocol requires

- a sufficiently large number of test instances,
- a predefined number (or proportion) of rejections.

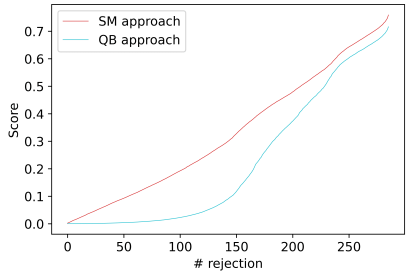
Threshold-based rejection protocol

- requires a predefined threshold on uncertainty score (↑),
- rejects instances whose scores are lower than the threshold.

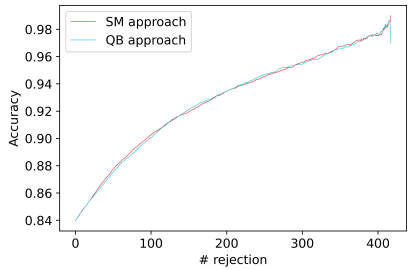
Results on noisy data sets [4]



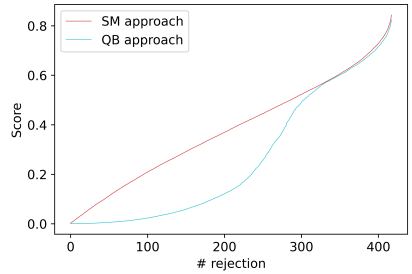
(a) derma. + SM



(b) derma. + SM



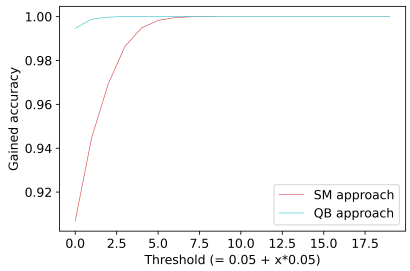
(c) forest + SM



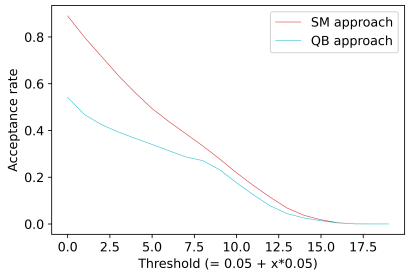
(d) forest + SM

Test accuracy and chosen score as the functions of the number of rejections

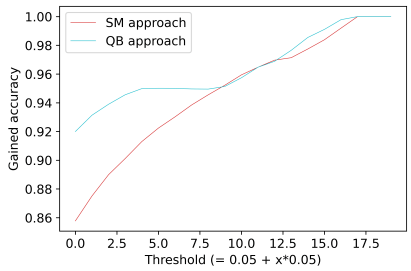
20×5 cross-validation with (train, test) = (20%, 80%)



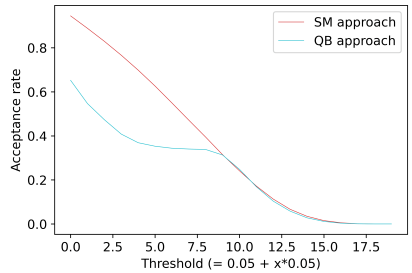
(a) derma. + SM



(b) derma. + SM



(c) forest + SM



(d) forest + SM

Test accuracy and acceptance rate as the functions of the threshold

20×5 cross-validation with (train, test) = (20%, 80%)

Experimental setting

Basic setting:

- Randomly flip the class of 25% of training + pool instances
- Using random forests of cardinality 100 as the ensembles
- The median P_{sE} is employed
- Assess smallest margin S_{SM} (↑) and $\mathbf{RS}_{\text{SM}}$ (↑)

Experimental setting

Basic setting:

- Randomly flip the class of 25% of training + pool instances
- Using random forests of cardinality 100 as the ensembles
- The median P_{sE} is employed
- Assess smallest margin S_{SM} (↑) and $\mathbf{RS}_{SM}$ (↑)

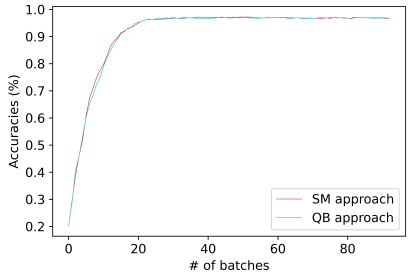
Budget based sampling protocol

- requires a predefined number (or proportion) of queries,
- stops when the predefined number is reached.

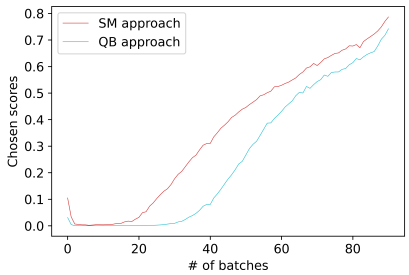
Threshold-based sampling protocol

- requires a predefined threshold on uncertainty score (↑),
- stops when the predefined threshold is reached.

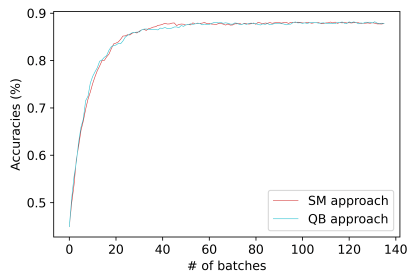
Results on noisy data sets [4]



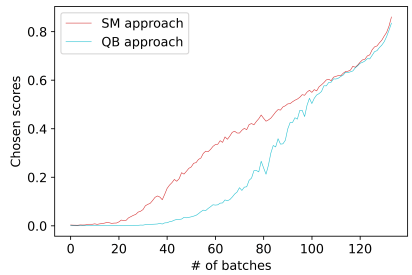
(a) derma. + SM



(b) derma. + SM



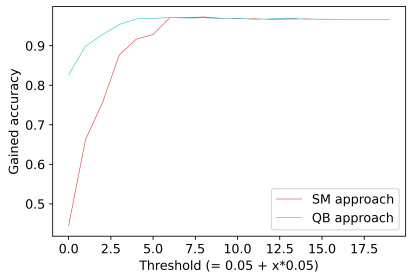
(c) forest + SM



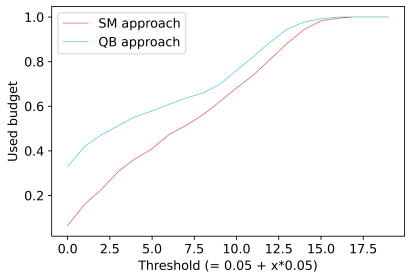
(d) forest + SM

Test accuracy and chosen score as the functions of the number of queries

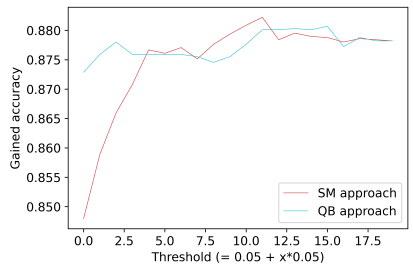
10×5 cross-validation with (train, pool, test) = (3%, 77%, 20%)



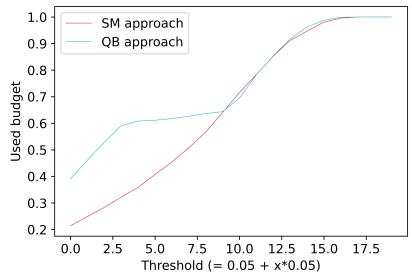
(a) derma. + SM



(b) derma. + SM



(e) forest + SM



(f) forest + SM

Test accuracy and used budget as the functions of the threshold

10×5 cross-validation with (train, pool, test) = (3%, 77%, 20%)

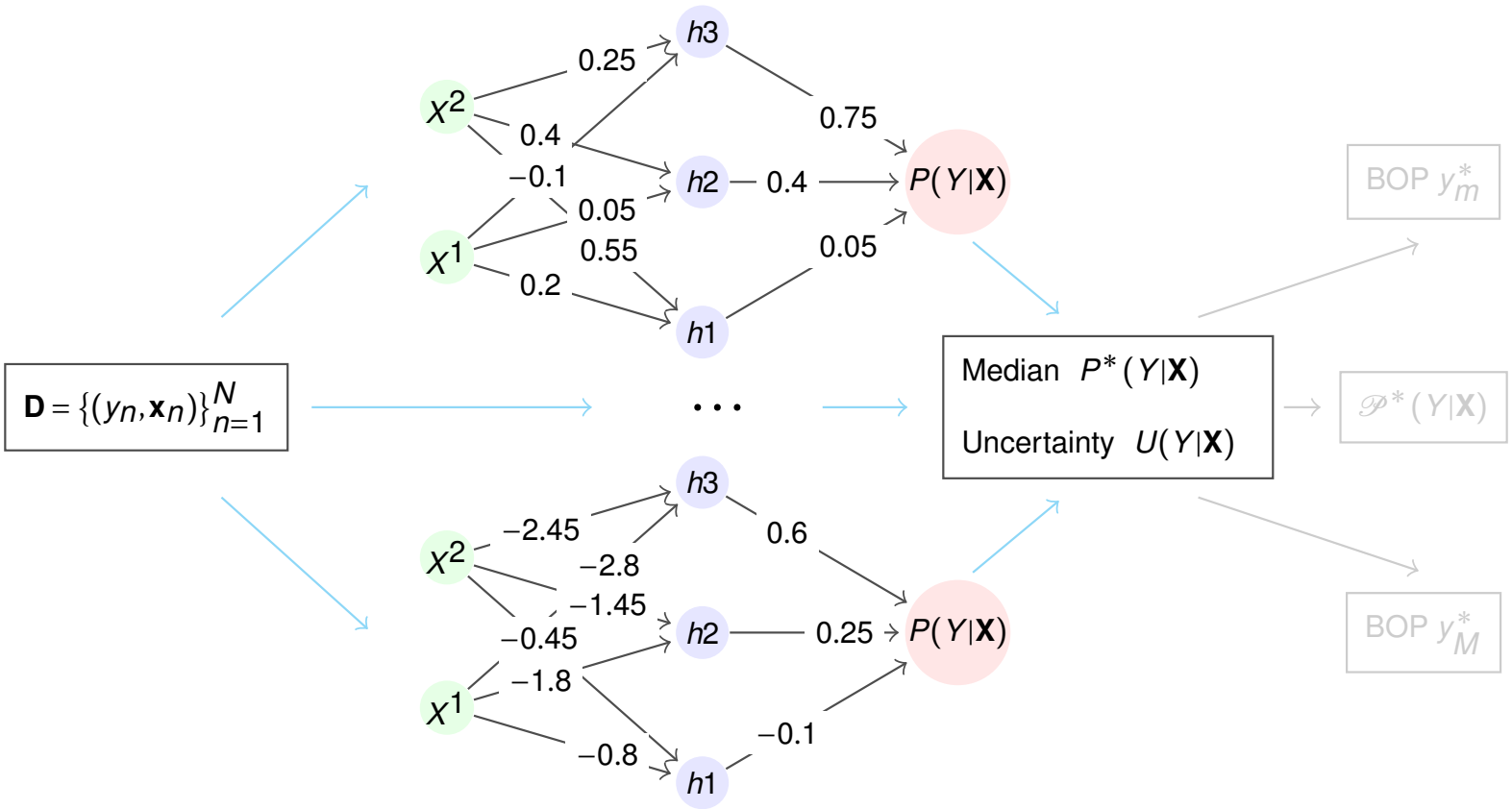
Outline

- Credal ensembling in multi-class classification
- Compact deep ensembles
  - Median and credal classifiers
  - Optimal Decision Rules
- Other applications

Outline

- Credal ensembling in multi-class classification
- Compact deep ensembles
  - Median and credal classifiers
  - Optimal Decision Rules
- Other applications

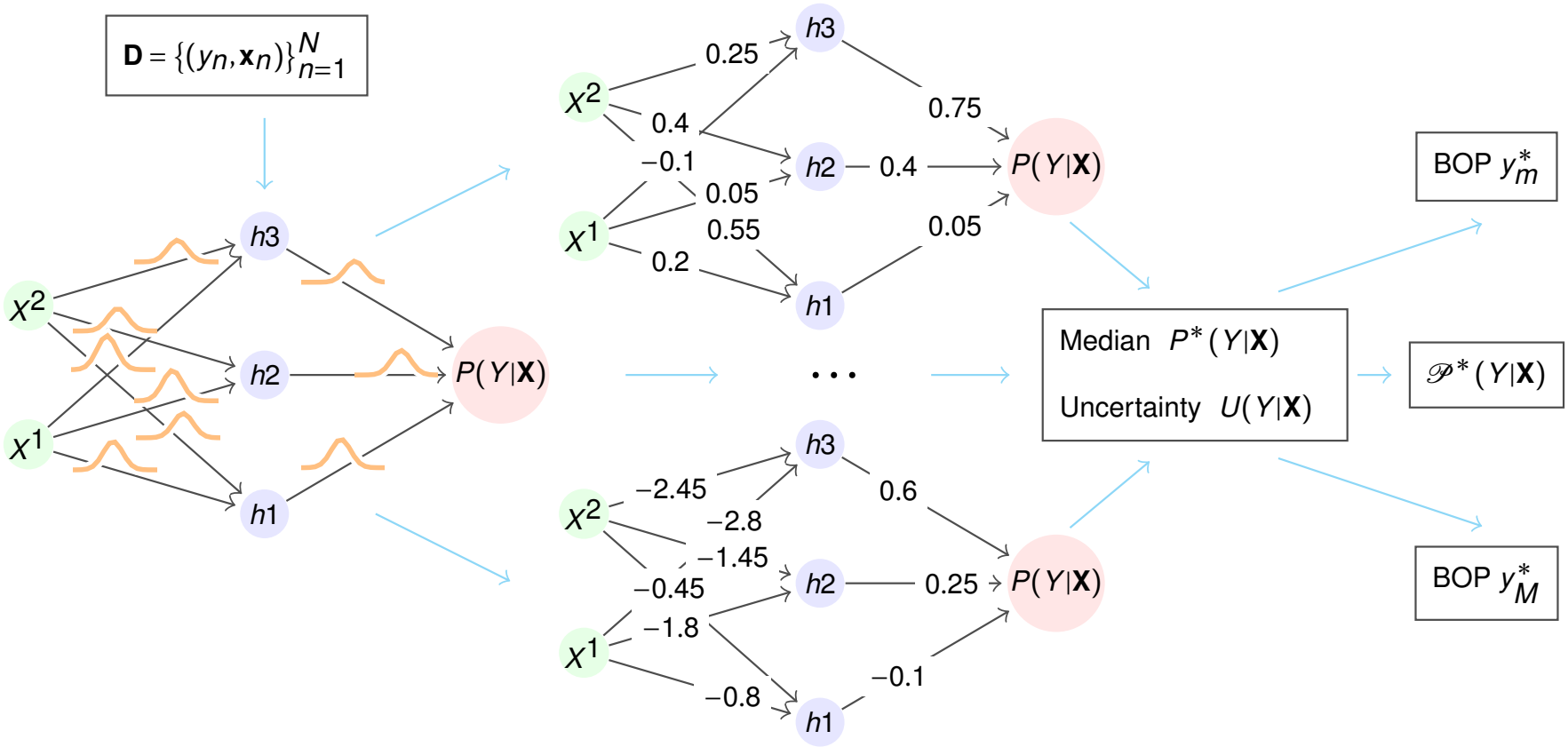
Conventional deep ensembles



Compared to the use of a single network:

- Much longer training time + Much larger storage memory
- Longer inference time

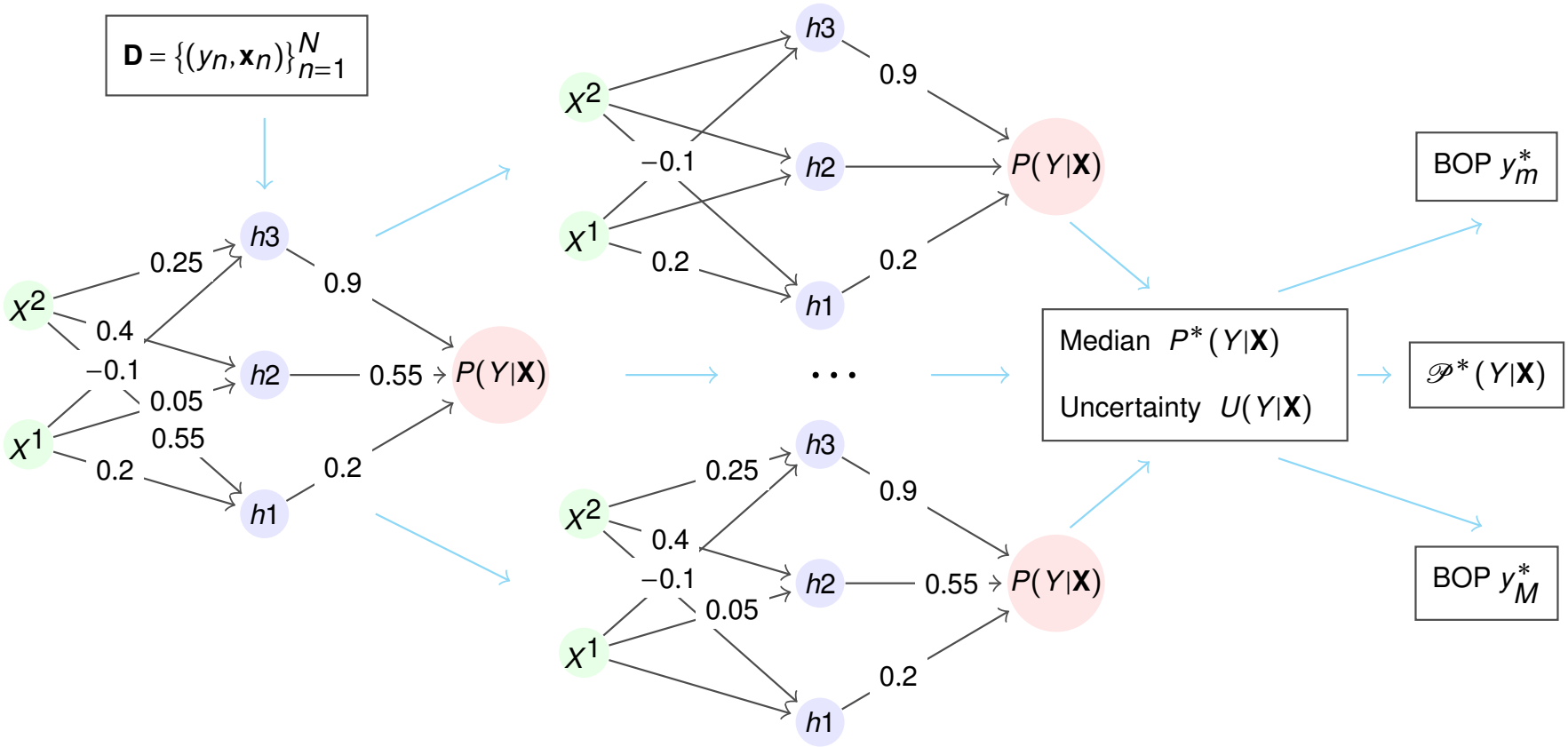
A BNN as an ensemble [7, 8]



Compared to the use of a single network:

- A bit longer training time + A bit larger storage memory
- Longer inference time

A CNN with dropout predictions as an ensemble [8]



Compared to the use of a single network:

- Similar training time + Similar storage memory
- Longer inference time

Outline

- Credal ensembling in multi-class classification
- Compact deep ensembles
  - Median and credal classifiers
  - Optimal Decision Rules
- Other applications

Probabilistic approach

Basic set (instance-wise manner):

- The median distribution P_d is given.

A **Bayes-optimal prediction (BOP) of** u is

$$y_u^{P_d} \in \operatorname*{argmax}_{y' \in \mathcal{Y}} \mathbf{E}\left[u(y', y)\right] = \operatorname*{argmax}_{y' \in \mathcal{Y}} \sum_{y \in \mathcal{Y}} u(y', y) P_d(y)\,. \tag{10}$$

Probabilistic approach

Basic set (instance-wise manner):

- The median distribution P_d is given.

A **Bayes-optimal prediction (BOP) of** u is

$$y_u^{P_d} \in \operatorname*{argmax}_{y' \in \mathcal{Y}} \mathbf{E}[u(y', y)] = \operatorname*{argmax}_{y' \in \mathcal{Y}} \sum_{y \in \mathcal{Y}} u(y', y) P_d(y). \tag{10}$$

A **Bayes-optimal prediction (BOP) of** U is

$$S_U^{P_d} \in \operatorname*{argmax}_{S' \subset \mathcal{Y}} \mathbf{E}[U(S', y)] = \operatorname*{argmax}_{S' \subset \mathcal{Y}} \sum_{y \in \mathcal{Y}} U(S', y) P_d(y). \tag{11}$$

Probabilistic approach

Basic set (instance-wise manner):

- The median distribution P_d is given.

A **Bayes-optimal prediction (BOP) of** u is

$$y_u^{P_d} \in \operatorname*{argmax}_{y' \in \mathcal{Y}} \mathbf{E}\left[u(y', y)\right] = \operatorname*{argmax}_{y' \in \mathcal{Y}} \sum_{y \in \mathcal{Y}} u(y', y) P_d(y) . \tag{10}$$

A **Bayes-optimal prediction (BOP) of** U is

$$S_U^{P_d} \in \operatorname*{argmax}_{S' \subset \mathcal{Y}} \mathbf{E}\left[U(S', y)\right] = \operatorname*{argmax}_{S' \subset \mathcal{Y}} \sum_{y \in \mathcal{Y}} U(S', y) P_d(y) . \tag{11}$$

Commonly used utilities do not account for reward sensitivity, e.g., utility-discounted accuracies:

$$U(S', y) = \frac{1}{g(|S'|)} [\![y \in S']\!] . \tag{12}$$

Probabilistic approach with reward sensitivity

Computational aspects [9]:

- How to define the higher the better utility $U : \mathscr{Y} \times 2^{\mathscr{Y}} \longmapsto \mathbb{R}_+$?
- How to efficiently compute BOP

$$S_U^{P_d} \in \underset{S' \subset \mathscr{Y}}{\operatorname{argmax}} \mathbf{E}\left[U(S', y)\right] = \underset{S' \subset \mathscr{Y}}{\operatorname{argmax}} \sum_{y \in \mathscr{Y}} U(S', y) P_d(y). \tag{13}$$

Probabilistic approach with reward sensitivity

Computational aspects [9]:

- How to define the higher the better utility $U : \mathscr{Y} \times 2^{\mathscr{Y}} \longmapsto \mathbb{R}_+$?
- How to efficiently compute BOP

$$S_U^{P_d} \in \underset{S' \subset \mathscr{Y}}{\operatorname{argmax}} \mathbf{E}\left[U(S', y)\right] = \underset{S' \subset \mathscr{Y}}{\operatorname{argmax}} \sum_{y \in \mathscr{Y}} U(S', y) P_d(y). \tag{13}$$

Desirable properties [9]:

- Normalization, universal feasibility
- Nested optimality
- Unimodality properties

Probabilistic approach with reward sensitivity (cont.)

Computational aspects [8]:

- $U : \mathscr{Y} \times 2^{\mathscr{Y}} \longmapsto \mathbb{R}_+$ is defined based on a u + a $g(|S'|)$:

$$U(S', y) = \frac{1}{g(|S'|)} \sum_{y' \in S'} u(y', y). \tag{14}$$

- The computation of BOP

$$S_U^{P_d} \in \underset{S' \subset \mathscr{Y}}{\operatorname{argmax}} \mathbf{E}\left[U(S', y)\right] = \underset{S' \subset \mathscr{Y}}{\operatorname{argmax}} \sum_{y \in \mathscr{Y}} U(S', y) P_d(y) \tag{15}$$

can be done in $O(|\mathscr{Y}|^2)$.

Probabilistic approach with reward sensitivity (cont.)

Computational aspects [8]:

- $U : \mathscr{Y} \times 2^{\mathscr{Y}} \longmapsto \mathbb{R}_+$ is defined based on a u + a $g(|S'|)$:

$$U(S', y) = \frac{1}{g(|S'|)} \sum_{y' \in S'} u(y', y). \tag{14}$$

- The computation of BOP

$$S_U^{P_d} \in \underset{S' \subset \mathscr{Y}}{\operatorname{argmax}} \mathbf{E}[U(S', y)] = \underset{S' \subset \mathscr{Y}}{\operatorname{argmax}} \sum_{y \in \mathscr{Y}} U(S', y) P_d(y) \tag{15}$$

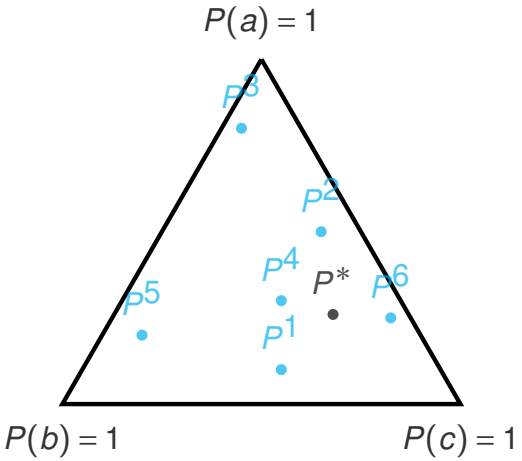
  can be done in $O(|\mathscr{Y}|^2)$.

Desirable properties [8]:

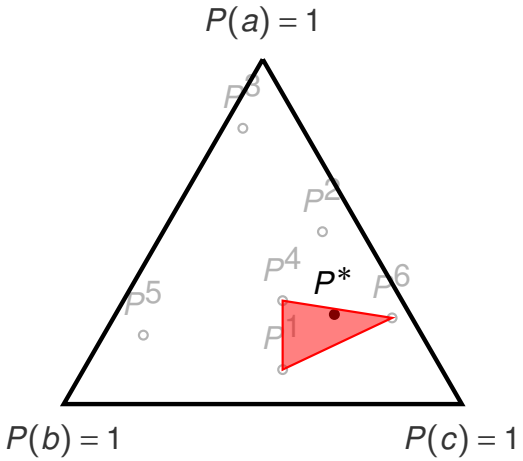
- Desirable properties discussed in [9] are satisfied.
- Monotonicity in recall, set-size, ... when changing a single parameter

Credal approach

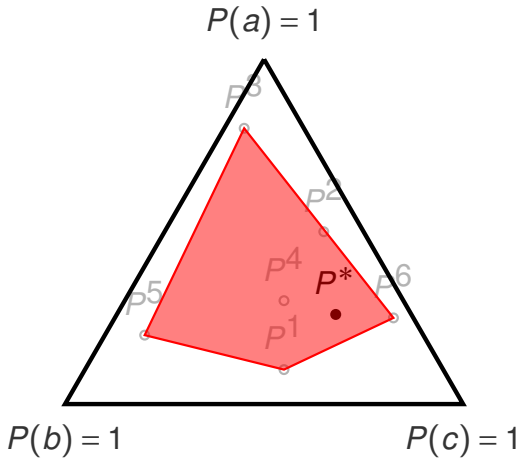
Basic set (instance-wise manner): The credal set $\mathbf{CH}^{d}_{\alpha^*}$ is given.



Ensemble **H** and P_{sE}



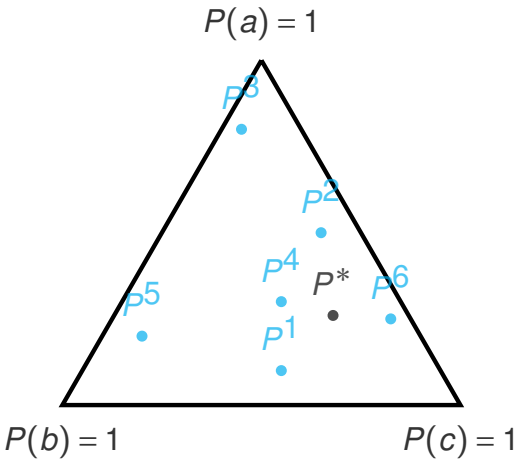
Credal set $\mathbf{CH}^{\text{sE}}_{0.5}$



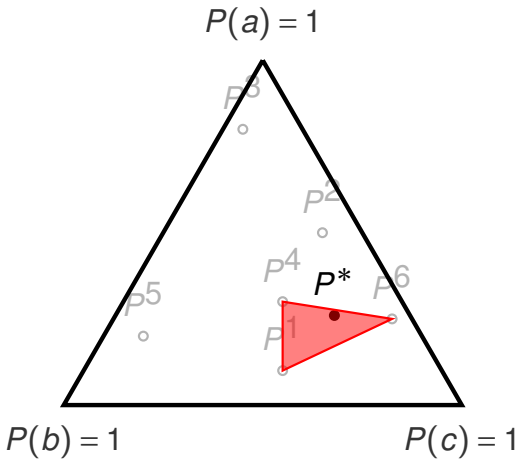
Credal set $\mathbf{CH}^{\text{sE}}_{0}$

Credal approach

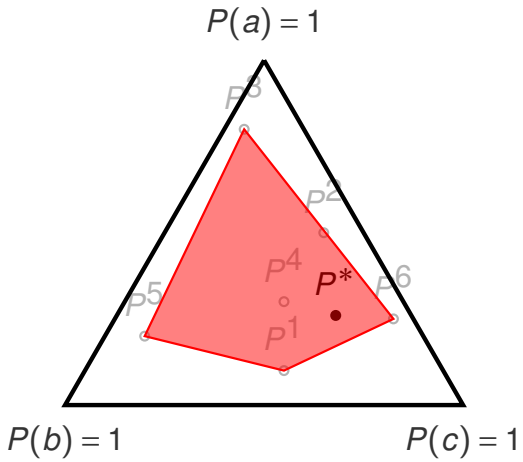
Basic set (instance-wise manner): The credal set $\mathbf{CH}^{d}_{\alpha^*}$ is given.



Ensemble $\mathbf{H}$ and P_{sE}



Credal set $\mathbf{CH}^{\text{sE}}_{0.5}$



Credal set $\mathbf{CH}^{\text{sE}}_{0}$

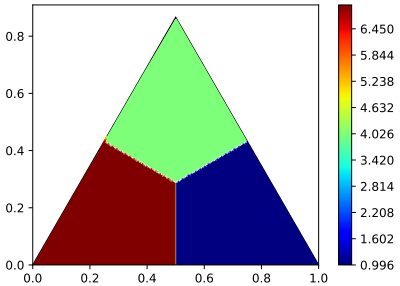
- Any IP decision rule $R_{\text{IP}} : 2^{\Delta^K} \longmapsto 2^{\mathcal{Y}}$ can be applied.
- Any related algorithmic solutions can be leveraged.

Outline

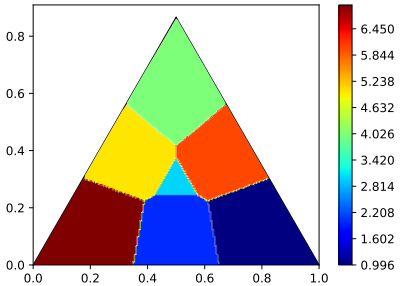
- Credal ensembling in multi-class classification
- Compact deep ensembles
- Other applications
  - Large language models with safety requirements
  - Credal Predictions for Out-of-distribution detection

Optimal Decision Rules

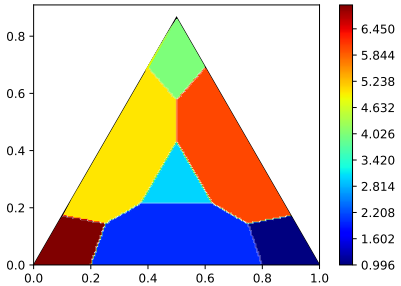
Probabilistic approach



0/1 loss

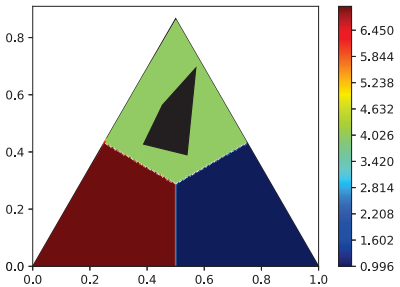


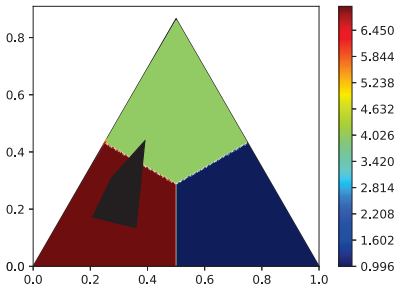
$u_{1.6}$

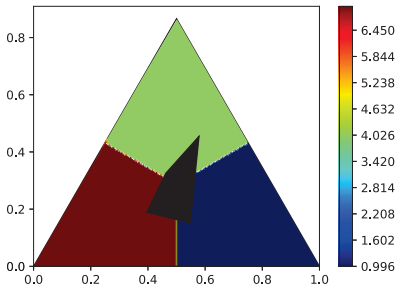


$u_{2.2}$

Credal approach







Outline

- Credal ensembling in multi-class classification
- Compact deep ensembles
- Other applications
  - Large language models with safety requirements
  - Credal Predictions for Out-of-distribution detection

Basic setup [3]

Multiple-choice question answering:

- Each prompt $\boldsymbol{x}$ is associated with a response $r_k \in \mathcal{Y} = \mathcal{R} = \{r_1, \ldots, r_K\}$.
- Predict the probability masses $P(r_k|\boldsymbol{x})$ $k = 1, \ldots, K$.
- Return a **Bayes-optimal prediction (BOP) of** u

$$r^u \in \operatorname*{argmax}_{r' \in \mathcal{Y}} \mathbf{E}\left[u(r', r)\right] = \operatorname*{argmax}_{y' \in \mathcal{Y}} \sum_{r \in \mathcal{Y}} u(r', r) P(r|\boldsymbol{x}).$$

Basic setup [3]

Multiple-choice question answering:

- Each prompt $\boldsymbol{x}$ is associated with a response $r_k \in \mathcal{Y} = \mathcal{R} = \{r_1, \ldots, r_K\}$.
- Predict the probability masses $P(r_k|\boldsymbol{x})$ $k = 1, \ldots, K$.
- Return a **Bayes-optimal prediction (BOP) of** u

$$r^u \in \operatorname*{argmax}_{r' \in \mathcal{Y}} \mathbf{E}\left[u(r', r)\right] = \operatorname*{argmax}_{y' \in \mathcal{Y}} \sum_{r \in \mathcal{Y}} u(r', r) P(r|\boldsymbol{x}).$$

Helpfulness and safety scores are given for each prompt:

- A helpfulness score $h_k(\uparrow)$, measuring how well r_k answers the query
- A safetyrisk score $s_k(\downarrow)$, measuring the likelihood that r_k violates safety policies

Basic setup [3]

Multiple-choice question answering:

- Each prompt $\boldsymbol{x}$ is associated with a response $r_k \in \mathcal{Y} = \mathcal{R} = \{r_1, \ldots, r_K\}$.
- Predict the probability masses $P(r_k|\boldsymbol{x})$ $k = 1, \ldots, K$.
- Return a **Bayes-optimal prediction (BOP) of** u

$$r^u \in \operatorname*{argmax}_{r' \in \mathcal{Y}} \mathbf{E}\left[u(r', r)\right] = \operatorname*{argmax}_{y' \in \mathcal{Y}} \sum_{r \in \mathcal{Y}} u(r', r) P(r|\boldsymbol{x}).$$

Helpfulness and safety scores are given for each prompt:

- A helpfulness score $h_k(\uparrow)$, measuring how well r_k answers the query
- A safetyrisk score $s_k(\downarrow)$, measuring the likelihood that r_k violates safety policies
- Expose the helpfulness-safety trade-off when predicting $P(r_k|\boldsymbol{x})$ $k = 1, \ldots, K$.

Helpfulness and safety scores

Let $r_1 \in \mathscr{Y} = \mathscr{R} = \{r_1, \ldots, r_K\}$ be a special safe fallback answer:

- r_1 is anodyne and policycompliant (for example, a refusal or a generic safe statement).
- r_1 contains 0 useful information ($h_k(\uparrow)$) but also incurs 0 risk $s_k(\downarrow)$.

Helpfulness and safety scores

Let $r_1 \in \mathscr{Y} = \mathscr{R} = \{r_1, \ldots, r_K\}$ be a special safe fallback answer:

- r_1 is anodyne and policycompliant (for example, a refusal or a generic safe statement).
- r_1 contains 0 useful information ($h_k(\uparrow)$) but also incurs 0 risk $s_k(\downarrow)$.

For each $r_k \in \mathscr{Y} = \mathscr{R}$, we can define

- The helpfulness lift ($\uparrow$): $H_k = h_k - h_1$
- The extra risk ($\downarrow$): $S_k = s_k - s_1$

Helpfulness and safety scores

Let $r_1 \in \mathscr{Y} = \mathscr{R} = \{r_1, \ldots, r_K\}$ be a special safe fallback answer:

- r_1 is anodyne and policycompliant (for example, a refusal or a generic safe statement).
- r_1 contains 0 useful information ($h_k(\uparrow)$) but also incurs 0 risk $s_k(\downarrow)$.

For each $r_k \in \mathscr{Y} = \mathscr{R}$, we can define

- The helpfulness lift ($\uparrow$): $H_k = h_k - h_1$
- The extra risk ($\downarrow$): $S_k = s_k - s_1$

For each probability distribution $P(\cdot|\boldsymbol{x})$, we define

$$\text{expected helpfulness lift} \quad \sum_{r_k \in \mathscr{Y}} H_k P(r_k|\boldsymbol{x}) \tag{16}$$

$$\text{expected extra risk} \quad \sum_{r_k \in \mathscr{Y}} S_k P(r_k|\boldsymbol{x}) \tag{17}$$

Constrained optimization problem

Predict $P(\cdot|\boldsymbol{x})$ to maximize expected helpfulness lift with acceptable expected risk

$$P \in \underset{P \in H^K}{\operatorname{argmax}} \sum_{r_k \in \mathcal{Y}} H_k P(r_k|\boldsymbol{x}) \tag{18}$$

$$\text{s.t} \sum_{r_k \in \mathcal{Y}} S_k P(r_k|\boldsymbol{x}) \leq T\,. \tag{19}$$

A more elaborate version + computational aspects can be found in [3].

Constrained optimization problem

Predict $P(\cdot|\boldsymbol{x})$ to maximize expected helpfulness lift with acceptable expected risk

$$P \in \underset{P \in H^K}{\operatorname{argmax}} \sum_{r_k \in \mathcal{Y}} H_k P(r_k|\boldsymbol{x}) \tag{18}$$

$$\text{s.t} \sum_{r_k \in \mathcal{Y}} S_k P(r_k|\boldsymbol{x}) \leq T\,. \tag{19}$$

A more elaborate version + computational aspects can be found in [3].

Probe classifiers

For each prompt $\boldsymbol{x}$ and each response $r_k \neq r_1$,

- Query helpfulness probe with question “Is this answer helpful? (Yes/No)" on $(\boldsymbol{x}, r_k)$ to obtain

$$h_k = \log\left(\frac{p_h^{\text{yes}}}{p_h^{\text{yes}} + p_h^{\text{no}}}\right). \tag{20}$$

Probe classifiers

For each prompt $\boldsymbol{x}$ and each response $r_k \neq r_1$,

- Query helpfulness probe with question “Is this answer helpful? (Yes/No)" on $(\boldsymbol{x}, r_k)$ to obtain

$$h_k = \log\left(\frac{p_h^{\text{yes}}}{p_h^{\text{yes}} + p_h^{\text{no}}}\right). \tag{20}$$

- Query safety probe with question “Is this answer risky? (Yes/No)" $(\boldsymbol{x}, r_k)$ to obtain

$$s_k = \log\left(\frac{p_s^{\text{yes}}}{p_s^{\text{yes}} + p_s^{\text{no}}}\right). \tag{21}$$

Probe classifiers

For each prompt $\boldsymbol{x}$ and each response $r_k \neq r_1$,

- Query helpfulness probe with question “Is this answer helpful? (Yes/No)" on $(\boldsymbol{x}, r_k)$ to obtain

$$h_k = \log\left(\frac{p_h^{\text{yes}}}{p_h^{\text{yes}} + p_h^{\text{no}}}\right). \tag{20}$$

- Query safety probe with question “Is this answer risky? (Yes/No)" $(\boldsymbol{x}, r_k)$ to obtain

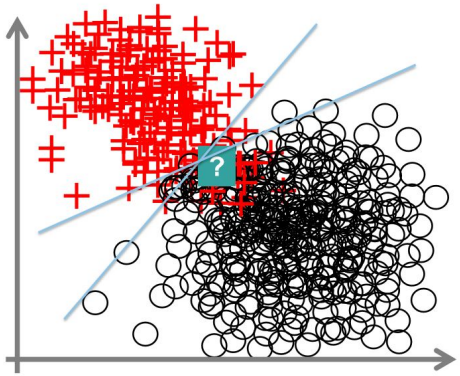
$$s_k = \log\left(\frac{p_s^{\text{yes}}}{p_s^{\text{yes}} + p_s^{\text{no}}}\right). \tag{21}$$

Probe classifiers can be LLM models [3].

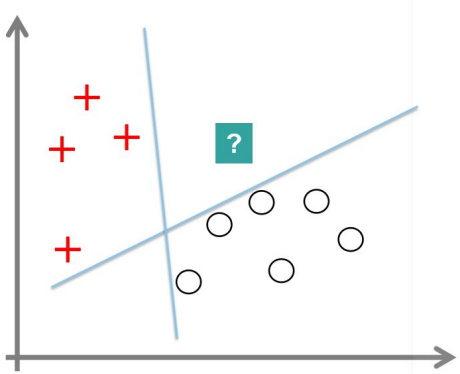
Outline

- Credal ensembling in multi-class classification
- Compact deep ensembles
- Other applications
  - Large language models with safety requirements
  - Credal Predictions for Out-of-distribution detection

Epistemic and aleatoric uncertainties [1, 6]

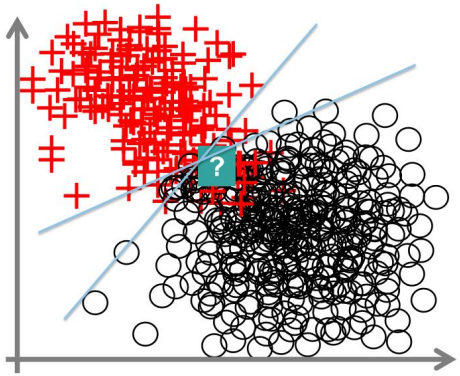


Aleatoric uncertainty: Classes are really mixed,
→ irreducible by collecting more information

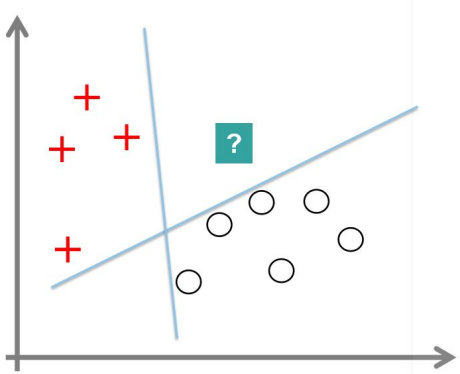


Epistemic uncertainty: Lack of information,
→ reducible by collecting more information

Epistemic and aleatoric uncertainties [1, 6]



Aleatoric uncertainty: Classes are really mixed,
→ irreducible by collecting more information



Epistemic uncertainty: Lack of information,
→ reducible by collecting more information

- Which kind of uncertainty is more interesting in OoD?
- How to quantify these degrees of uncertainty?

Basic setup

The given training data $\boldsymbol{D} \subset \mathscr{X} \times \mathscr{Y}$

- is used to estimate a classifier,
- which predicts, for each instance $\boldsymbol{x}$, $\mathscr{P}(Y|\boldsymbol{x})$.

Basic setup

The given training data $\boldsymbol{D} \subset \mathscr{X} \times \mathscr{Y}$

- is used to estimate a classifier,
- which predicts, for each instance $\boldsymbol{x}$, $\mathscr{P}(Y|\boldsymbol{x})$.

$\mathscr{P}(Y|\boldsymbol{x})$ is used to quantify the degree of epistemic uncertainty $\mathrm{EU}(\boldsymbol{x})$:

- using its volume (not recommended!) [5].
- using the disconsensus of its members [2].

Basic setup

The given training data $\boldsymbol{D} \subset \mathscr{X} \times \mathscr{Y}$

- is used to estimate a classifier,
- which predicts, for each instance $\boldsymbol{x}$, $\mathscr{P}(Y|\boldsymbol{x})$.

$\mathscr{P}(Y|\boldsymbol{x})$ is used to quantify the degree of epistemic uncertainty $\mathrm{EU}(\boldsymbol{x})$:

- using its volume (not recommended!) [5].
- using the disconsensus of its members [2].

Prediction: Instances with high $\mathrm{EU}(\boldsymbol{x})$ are predicted as OoD instances.

An (approximate) detail framework [2]

The degree of epistemic uncertainty on $\mathscr{P}(Y|\boldsymbol{x})$

$$\mathrm{EU}(\boldsymbol{x}) = \overline{U}_E(\boldsymbol{x}) - \underline{U}_E(\boldsymbol{x}), \tag{22}$$

where U_E is the entropy (see Lecture 2) and

$$\overline{U}_E(\boldsymbol{x}) = \max_{P \in \mathscr{P}(Y|\boldsymbol{x})} U_E(P(Y|\boldsymbol{x}), \tag{23}$$

$$\underline{U}_E(\boldsymbol{x}) = \min_{P \in \mathscr{P}(Y|\boldsymbol{x})} U_E(P(Y|\boldsymbol{x}) \tag{24}$$

An (approximate) detail framework [2]

The degree of epistemic uncertainty on $\mathscr{P}(Y|\boldsymbol{x})$

$$\mathrm{EU}(\boldsymbol{x}) = \overline{U}_E(\boldsymbol{x}) - \underline{U}_E(\boldsymbol{x}), \tag{22}$$

where U_E is the entropy (see Lecture 2) and

$$\overline{U}_E(\boldsymbol{x}) = \max_{P \in \mathscr{P}(Y|\boldsymbol{x})} U_E(P(Y|\boldsymbol{x}), \tag{23}$$

$$\underline{U}_E(\boldsymbol{x}) = \min_{P \in \mathscr{P}(Y|\boldsymbol{x})} U_E(P(Y|\boldsymbol{x}) \tag{24}$$

$\mathscr{P}(Y|\boldsymbol{x})$ is estimated using interval probabilities (See lecture 2)

$$\left[\underline{P}(y|\boldsymbol{x}), \overline{P}(y|\boldsymbol{x})\right], \forall y \in \mathscr{Y}. \tag{25}$$

References I

[1] E. Hüllermeier and W. Waegeman.
Aleatoric and epistemic uncertainty in machine learning: An introduction to concepts and methods.
Machine learning, 110(3):457–506, 2021.

[2] T. Löhr, P. Hofman, F. Mohr, and E. Hüllermeier.
Credal prediction based on relative likelihood.
arXiv preprint arXiv:2505.22332, 2025.

[3] T. Nguyen and L. Tran-Thanh.
Safety game: Balancing safe and informative conversations with blackbox agentic ai using lp solvers.
arXiv preprint arXiv:2510.09330, 2025.

[4] V.-L. Nguyen, H. Zhang, and S. Destercke.
Credal ensembling in multi-class classification.
Machine Learning, pages 1–64, 2024.

[5] Y. Sale, M. Caprio, and E. Hüllermeier.
Is the volume of a credal set a good measure for epistemic uncertainty?
In *Proceedings of the Thirty-Ninth Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 1795–1804, 2023.

[6] R. Senge, S. Bösner, K. Dembczyński, J. Haasenritter, O. Hirsch, N. Donner-Banzhoff, and E. Hüllermeier.
Reliable classification: Learning classifiers that distinguish aleatoric and epistemic uncertainty.
Information Sciences, 255:16–29, 2014.

[7] K.-D. Tran, X.-T. Hoang, V.-L. Nguyen, S. Destercke, and V.-N. Huynh.
Robust classification in bayesian neural networks.
In *Proceedings of the Eleventh International Symposium on Integrated Uncertainty in Knowledge Modelling and Decision Making (IUKM)*, pages 29–41. Springer, 2025.

References II

[8] K.-D. Tran, D.-M. Nguyen, V.-L. Nguyen, X.-T. Hoang, S. Destercke, and V.-N. Huynh.
Compact set-valued deep ensembling in multi-class classification.
arXiv preprint arXiv:2610.05332, 2026.

[9] G. Yang, S. Destercke, and M.-H. Masson.
The costs of indeterminacy: how to determine them?
IEEE transactions on cybernetics, 47(12):4316–4327, 2016.